The Thesis committee for Nam Phuong Pham
Certifies that this is the approved version of the following thesis:

Automatic channel detection using deep learning

APPROVED BY

SUPERVISING COMMITTEE:

Sergey B. Fomel, Supervisor

Mrinal K. Sen

Zoltan Sylvester

Automatic channel detection using deep learning

by

Nam Phuong Pham

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Geological Sciences

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2019

Dedicated to my parents who supported my education and provided encouragement over the years

Acknowledgments

I owe a great deal of gratitude to Dr. Sergey Fomel. He has provided me with the opportunity to continue my education in geophysics which is one of the most challenging, unique, and interesting fields of study. Sergey is open to any ideas that I come up with to solve problems. He encourages me to approach modern methods such as machine learning and deep learning for solving geophysical problems. I greatly appreciate everything Sergey has taught me during my time in Austin.

The work discussed in this thesis is mainly from Pham et al. (2018, 2019). I am very thankful to the co-authors on this work in particular for their assistance and input on the project.

The Texas Consortium for Computation Seismology (TCCS) has been home to me for the last two years, and the members of this group are kind, supportive and hard working. I am very thankful for past and present members of the group that I have the opportunity to work with: Dr. Xinming Wu, Dr. Zhiguang Xue, Dr. Yanadet Sripanich, Dr. Dmitrii Merzlikin, Yunzhi Shi, Mason Phillips, Luke Decker, Yunan Yang, Ben Gremillion, Zhicheng Geng, Harpreet Kaur, Sarah Greer, and Sean Bader. I also appreciate the financial support and insights provided by the sponsors of TCCS. I am also thankful to the developers of `Madagascar` open-source software package and `Tensorflow` open-source machine learning framework, where I did all of the computations in this thesis.

Additionally, I would like to thank Dallas Dunlap and Dr. Jacob Covault for many helpful discussions and providing a geologist view of channel detection problem.

I also would like to thank Philip Guerrero who supports all of the geoscience graduate students at UT Austin. I appreciate the members of my committee, Dr. Mrinal Sen and Dr. Zoltan Sylvester for their feedback and recommendations on the work presented in this thesis.

Also, I am thankful to many people whom I have had the opportunity to interact and work with since I started out at The University of Tulsa, OK. During my undergraduate studies, I was lucky to be surrounded by a group of friends and professors who are passionate about geology and geophysics and driven to succeed. I am especially thankful to Dr. Jingyi Chen for introducing me to the field of geophysics and guiding me to explore machine learning. Also, I want to thank Dr. Bryan Tapp and Dr. Kumar Ramachandran who introduced me to different geologic concepts and problems.

I am very thankful to my parents who always support and encourage me in spite of long geographical distance.

Automatic channel detection using deep learning

Nam Phuong Pham, M.S.Geo.Sci.

The University of Texas at Austin, 2019

Supervisor: Sergey B. Fomel

Picking 3D channel geobodies in seismic volumes is an important objective in seismic interpretation for hydrocarbon exploration. Manual detection of channel geobodies is a time-consuming and subjective process. The interpreter can calculate different seismic attributes such as coherence to aid for manual detection of channel geobodies in seismic volumes. However, these attributes still do not directly identify 3D channel geobodies.

Machine learning and deep learning are data-driven techniques that have been getting more attention recently in different fields, such as medical imaging and computer vision. With large volumes of available data in different types and a development of powerful computational resources, geophysics is a promising field for applying machine learning and deep learning. Many seismic interpretation steps are analogous to different problems in computer vision that have been solved successfully using deep learning. Channel detection in seismic volumes is analogous to segmentation problems for images. Applying deep learning to seismic interpretations, specifically

to automatic channel detection in 3D seismic volumes, can make the process faster and the workflow less subjective. Decision-making based on interpretations is uncertain; so uncertainties in interpretation results are very important. Deep learning with different algorithms can also help interpreters quantify this uncertainty.

Table of Contents

List of Figures

Chapter 1

Introduction

Channels, especially deep-water channels, are important geologic features in oil and gas exploration. Water and sediment flow through these linear, commonly concave-based depression structures and are deposited to form reservoir targets. The close proximity of coarse-grained and fine-grained sediments in some channels can create stratigraphic hydrocarbon traps. Geologists interpret channels in seismic volumes by picking top and base horizons then filling the sand volumes inside. The interpretations can take lots of time for big 3D seismic volumes, can be biased, and hard to quantify the uncertainty. The development of methods that automatically and efficiently detect channel features in seismic volumes and quantify the detection uncertainty is a key to speed-up seismic interpretation process in oil and gas exploration. My research focuses on developing a method based on an enoder-decoder convolutional neural network that can help interpreters detect channel bodies in 3D seismic volumes rapidly and efficiently with quantified uncertainty.

Machine learning and deep learning have become very popular in computer science and are used widely in different areas such as medical imaging and oil and gas industry. The large amount of data and development of computing resources such as parallel computing, graphics processing units (GPUs), and tensor processing units (TPUs) allow machine learning to be used effectively. Geophysics involves

analysis of a large amount of different data from seismic, well-log, gravity, GPR, and magnetic data. Some of these data have been processed and interpreted by specialists, and are excellent resources for machine learning and deep learning to automatize interpretations in oil and gas exploration effectively and fast. Computers can learn from old data and apply to new data. Human can enhance computers' interpretations without starting from scratch. Channel detection in seismic can be automated using deep learning models with synthetic data created by geologists, geophysicists, and engineers.

In the first part of this thesis, I focus on an overview of machine learning, deep learning and image segmentation in computer vision. Machine learning and deep learning are fields of computer science that allow computers to "learn" and get information from the data. The computers then have an ability to predict on new data that they haven't seen during training. Many algorithms have been developed in machine learning and deep learning such as support vector machines, recurrent neural networks, and convolutional neural networks, which can be applied to solve different problems in computer vision and natural language processing. I focus on the image segmentation problem in computer vision with convolutional neural networks, which is a very popular research topic in computer science. Encoder-decoder convolutional neural networks are one method that has an encoder to automatically extract useful features from image and a decoder to upsample them back to original image size. Dropout layers can also be used in training and testing to learn the distribution of weights in convolutional layers and quantify the model uncertainty. This model succeeded in doing segmentation with different datasets (Badrinarayanan et al., 2015; Ronneberger et al., 2015). The channel detection problem in seismic can be defined as an image segmentation problem in which there are two labels of channel or non-

channel.

In the second part of this thesis, I discuss several methods for enhancing channel boundaries in seismic images such as coherence, Sobel filter, and methods to highlight channel geobodies such as sweetness and spectral decomposition. I propose an approach to detect channel geobodies in 3D seismic volumes automatically by utilizing an encoder-decoder convolutional neural network mentioned in the first part of this thesis. Synthetic model is created to simulate a stacked channel system, which is then used to train an encoder-decoder convolutional neural network. The trained weights are then applied directly to a field seismic data to detect channels automatically. A dropout layer is used in training and testing to quantify the uncertainty of the model. The uncertainty helps interpreters judge and enhance the result from deep learning algorithm. Unlike other conventional methods, my approach is not limited to enhance channel boundaries but can effectively detect channel geobodies in seismic volumes without having any human interpretations on seismic attributes. This approach is a novel workflow combining knowledge from experts and deep learning algorithms in automatic seismic interpretation for features detection.

THESIS OUTLINE

In Chapter 2, I discuss machine learning and deep learning in general. I review a novel deep learning architecture called convolutional neural networks including different layers such as convolutional layers, pooling layers, dropout layers, and activation layers. I also discuss how convolutional neural networks are viewed in a Bayesian perspective to understand the uncertainty. Finally, I introduce an encoder-decoder convolutional neural network called SegNet, which is used for image segmentation

problems in computer vision and serves as the key architecture for the method I propose in latter chapters.

In Chapter 3, I review some methods for enhancing channel boundaries in seismic images including coherence attribute and Sobel filter. I also mention other seismic attributes that are useful for channel detection in seismic images such as sweetness attribute and spectral decomposition. I propose a method using an encoder-decoder convolutional neural network described in Chapter 4 and Chapter 5 for automatic channel detection in 3D seismic volumes. My proposed method can pick not only channel edges but also 3D channel geobodies without any human-generated seismic attributes. None of the conventional methods can quantify the uncertainty of the detection model. Understanding the confidence with which we can trust the channel detection output is important for decision making in exploration. Together with the probability of channels' presence in 3D seismic volume, my method also produces the model uncertainty at every pixel.

In Chapter 4, I discuss the synthetic channels model that is used in training the encoder-decoder convolutional neural network mentioned in Chapter 2. I also discuss in detail different hyper-parameters used in training. Results after training are probability volume and model uncertainty volume. First volume shows probability of channel at every pixel in a 3D seismic volume. When expressing deep learning in a Bayesian way, the uncertainty of deep learning model comes from different sources such as parameters uncertainty or model structure uncertainty. In this thesis, the model uncertainty is only produced from the parameters uncertainty. The uncertainty volume quantifies how much we can trust the segmentation result at every pixel in a 3D seismic volume. The probability volume shows the efficiency of deep learning model in picking 3D channel geobodies in seismic volumes.

In Chapter 5, I use the trained model and weights from the previous chapters to automatically pick 3D channel geobodies in two 3D field datasets, from Browse Basin offshore Australia and from Parihaka New Zealand. The testing result is better if the field data has similar characteristics with the training data such as the offshore Australia dataset. However, the trained model is somehow generalized and shows good results in different channel system such as the Parihaka dataset.

In Chapter 6, I conclude this thesis with a brief summary and discussion of the results. I discuss the advantages and disadvantages of the method introduced in this thesis and consider potential future applications of this work in the seismic interpretation workflow.

Chapter 2

Deep Learning Review

In this chapter, I discuss several concepts used in machine learning and deep learning including different artificial intelligent methods for classification and segmentation problems. I also review common layers in convolutional neural networks which is one of the most popular method in deep learning. I also discuss how deep learning is expressed in a Bayesian way. Finally, I discuss in detail the SegNet, an encoder-decoder convolutional neural network architecture, which is the primary tool I use for automatic channel detection.

ARTIFICIAL INTELLIGENCE: MACHINE LEARNING AND DEEP LEARNING

Artificial intelligence (AI) is a field of computer science, which emphasizes on simulating intelligent machines that can think and react like human. AI researches started in the 1950s solving word problems in algebra, proving logical theorems, and teaching computers to learn checkers strategies (Samuel, 1959; McCorduck, 2004). In the 1960s and 1970s, the US Department of Defense was interested in AI and funded many researches in training computers to mimic basic human reasoning such as street mapping projects (McCorduck, 2004). These early works approached the problem with different symbolic methods and neural networks. They stirred the excitement for "thinking machines".

Machine learning (ML) is a branch of AI, which is based on the idea that machines can learn from data, identify patterns, and make predictions or decisions without being programmed to perform specific tasks (Koza et al., 1996). ML started to become popular in the 1990s. It shifted focus toward methods and models borrowed from statistics and probability theory such as support vector machine (SVM) and expectation-maximization (EM) algorithm (Langley, 2011). ML has become popular thanks to increased availability of digitized data volumes and advanced algorithms. ML methods are divided into supervised and unsupervised algorithms. Unsupervised ML methods only require data but supervised ML methods also require the corresponding labels. Data with labels is abundant in computer science such as famous ImageNet and COCO datasets with natural images (Lin et al., 2014; Russakovsky et al., 2015). However, in geophysics, these labeled datasets are not always available.

Deep learning (DL) is a type of ML, which is based on learning data representations. Traditional approaches of ML use engineer features to estimate the parameters of an analytic model. These methods depend on the quality of the model and handcrafted features which are not all meaningful and useful. DL replaces manual feature engineering, allows machines to both automatically discover the features from raw data and use them to perform a specific task such as classification and segmentation. DL also replaces the formulation of the model with hierarchical characterizations (layers) that learn to recognize features of data. It shifted from telling the computers how to solve a problem to training the computers to solve the problem themselves. DL was first introduced to the ML community in 1986 but it took long time to train a simple model. The "big bang" of deep learning started in 2009 as "deep learning neural networks were trained with Nvidia graphics processing units (GPUs)" (Nvidia CEO). DL is attracting more attention because of more powerful

computational resources and more data. Two powerful DL architectures are convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs are a specialization of the neural networks for data in the form of multiple arrays (LeCun et al., 2015). CNNs are used widely in computer vision for image segmentation and classification. RNNs, which were developed in the 1980s, are a class of artificial neural network where connections between nodes form a directed graph along a sequence. RNNs use their internal state to process sequences of inputs and exhibit temporal dynamic behavior for a time sequence. Therefore, RNNs are powerful for natural language processing tasks such as handwritting recognition and speech recognition. Two famous RNN models are long short-term memory (LSTM) and gated recurrent unit (GRU). They are used to avoid the vanishing and exploding gradient problem in RNNs by introducing recurrent gates, update gates, output gates, and forget gates (Hochreiter and Schmidhuber, 1995, 1997).

Traditional machine learning methods have been used for a long time in geophysics. Wang and Mendel (1992); Calderón-Macías et al. (1997) cast deconvolution and multiple attenuation problems as a Hopfield network. Hopfield network proposed by (Hopfield and Tank, 1985) is a physics-based single-layer recurrent network whose dynamics are controlled by a an energy function and a system of nonlinear ordinary differential equations. The energy function is calculated by the neuron states, connections between neurons, and externally supplied inputs to neurons. Statistical machine learning methods have been used for geophysical interpretations. SVM is used for AVO classification, faults identification (Guitton et al., 2017); and K-means clustering algorithm is used to detect salt boundary (Di et al., 2018). These traditional techniques do not require much data but attributes are created manually to be fed into the algorithms. Large number of data collected from different sources such

as well-log, seismic, gravity, electromagnetic and the development of more powerful computational resources are excellent conditions for applying deep learning in geophysics. Machines automatically extract features from data to perform interpretation tasks. Instead of manually extracting twelve seismic attributes in Di et al. (2018), Waldeland and Solberg (2017) and Shi et al. (2018) use only seismic amplitude as an input in a convolutional neural network for salt-body classification. Wu et al. (2019) also use convolutional neural network with seismic amplitude for faults detection. Encoder-decoder convolutional neural network is the main idea behind the method presented in this thesis for automatic 3D channel geobodies detection in seismic which was published in (Pham et al., 2018, 2019). The neural network automatically learns useful feature maps by different convolutional filters.

Traditional machine learning methods

Traditional ML algorithms are mainly based on statistics such as SVM and K-means. K-means is an unsupervised clustering algorithm that has two steps: clustering assignment and moving centroid step. It starts with an initial guess for centroids of the clusters and data is sorted into corresponding groups based on the distances to these centroids, then these centroids of clusters are updated slightly based on the means of clusters. K-nearest neighbor and SVM are supervised algorithms for classification problems. K-nearest neighbor algorithm finds top k closet training data points with labels to a test data point and has them vote on the label of the test point. It is an extension of the nearest neighbor classifier using commonly L1 and L2 distance

$$\begin{aligned} d_1(\mathbf{I}_1, \mathbf{I}_2) &= \sum_P |\mathbf{I}_1^P - \mathbf{I}_2^P|, \\ d_2(\mathbf{I}_1, \mathbf{I}_2) &= \sqrt{\sum_P (\mathbf{I}_1^P - \mathbf{I}_2^P)^2}, \end{aligned} \tag{2.1}$$

where $\mathbf{I}_i$ is a data point viewed as a p-dimensional vector. Linear SVM algorithm separates these data points with a (p-1)-dimensional hyperplane such that the distance from it to the nearest data point on each side is maximized. Boser et al. (1992) propose a nonlinear SVM classifier by applying the kernel trick to the original algorithm.

Artificial neural network (ANN) is a computational learning system that uses a network of functions to understand and translate data inputs to target outputs. This system learns to do tasks by considering examples, generally without being programmed with any task-specific rules. ANN has a collection of connected units, which loosely resemble the neurons in a biological brain (Figure 2.1). An artifical neuron receives a signal and then proceeds it by a set of weights, biases, and a nonlinear activation function

$$\mathbf{o}_j(t) = \sigma\left(\sum_i \mathbf{o}_i(t)\,\mathbf{w}_{i\,j} + \mathbf{b}_j\right), \tag{2.2}$$

where $\mathbf{o}_j(t)$ and $\mathbf{o}_i(t)$ are the states of neuron j and i depending on a discrete time paramater, $\mathbf{w}_{i\,j}$ is the connected weight between neuron i and j, and $\mathbf{b}_j$ is the bias. The weights and biases are updated during learning process. σ is a nonlinear activation function such as tanh, sigmoid, softmax, or rectified linear unit (ReLU). Artificial neurons are organized into layers performing different kinds of transformations on their inputs. ANN can also be viewed as a chain of operators. ANN is a mathematical model finding a function $f : \mathbf{X} \to \mathbf{Y}$. The function is obtained by varying parameters or architecture of the network. For $x \in \mathbf{X}$, function $f(x)$ is defined as a composition of other functions $g_i(x)$, which can further be decomposed into other

functions

$$f(x) = \sigma\left(\sum_i w_i\, g_i(x)\right),$$
$$g_i(x) = \sigma\left(\sum_j w_j\, h_j(x)\right). \tag{2.3}$$

The activation function introduces a nonlinear relationship into the network and provides a smooth transition as input values change. Activation functions constrain the output from a neuron to prevent exploding gradient. Each activation function is used for different purposes of ANNs. Tanh function makes the output between -1 and 1 ($\tanh(\mathbf{z}) = \frac{e^{\mathbf{z}}-e^{-\mathbf{z}}}{e^{\mathbf{z}}+e^{-\mathbf{z}}}$) and sigmoid function makes the output between 0 and 1 ($\sigma(\mathbf{z}) = \frac{e^{\mathbf{z}}}{e^{\mathbf{z}}+1}$) so it is used widely for binary classification problems where the outputs are probabilities (Figure 2.2). Another function useful for multi-class classification problems is the softmax function $s(\mathbf{z})_j = \frac{e^{\mathbf{z}}_j}{\sum_{k=1}^{K} e^{\mathbf{z}}_k}$ where $j = 1, ..., K$ and $\sum_{j=1}^{K} s(\mathbf{z})_j = 1$. ReLU function makes the output between 0 and ∞ ($f(\mathbf{z}) = \max(0, \mathbf{z})$) (Figure 2.3).

Given a specific task, the objective of learning process using a set of observations in neural networks is finding a function f^* mapping $\mathbf{X}$ to $\mathbf{Y}$, which minimizes a specific cost function C. The most important concept that makes neural networks popular is backpropagation (Werbos, 1974, 1994). The parameters of the networks can be updated using stochastic gradient descent

$$\mathbf{w}_{i\,j}(t+1) = \mathbf{w}_{i\,j}(t) - \eta\, \frac{\partial C}{\partial \mathbf{w}_{i\,j}}, \tag{2.4}$$

where η is the learning rate. Backpropagation calculates the gradient of the cost function with respect to the parameters of neural networks using chain rule. The state of a neuron j is calculated using equation 2.2 so the partial derivatives of cost function with respect to parameters $\mathbf{w}_{k\,j}$ and $\mathbf{b}_j$ are

$$\frac{\partial C}{\partial \mathbf{w}_{k\,j}} = \frac{\partial C}{\partial \mathbf{o}_j}\, \frac{\partial \mathbf{o}_j}{\partial \left(\sum_i \mathbf{o}_i(t)\, \mathbf{w}_{i\,j} + \mathbf{b}_j\right)}\, \frac{\partial \left(\sum_i \mathbf{o}_i(t)\, \mathbf{w}_{i\,j} + \mathbf{b}_j\right)}{\partial \mathbf{w}_{k\,j}}, \tag{2.5}$$

$$\frac{\partial C}{\partial \mathbf{b}_j} = \frac{\partial C}{\partial \mathbf{o}_j} \frac{\partial \mathbf{o}_j}{\partial \left(\sum_i \mathbf{o}_i(t)\, w_{i\,j} + \mathbf{b}_j\right)} \frac{\partial \left(\sum_i \mathbf{o}_i(t)\, \mathbf{w}_{i\,j} + \mathbf{b}_j\right)}{\partial \mathbf{b}_j}. \tag{2.6}$$

Let square error function be the cost function $C = \frac{1}{2}(\mathbf{t} - \mathbf{y})^2$ where $\mathbf{t}$ is the target output, $\mathbf{y}$ is the actual output of output neuron and sigmoid function be the nonlinear activation function $\sigma(\mathbf{z}) = \frac{1}{1+e^{-\mathbf{z}}}$ for example. Then the partial derivative of the output of neuron j with respect to its input is

$$\frac{\partial \mathbf{o}_j}{\partial \left(\sum_i \mathbf{o}_i(t)\, \mathbf{w}_{i\,j} + \mathbf{b}_j\right)} = \sigma\left(\sum_i \mathbf{o}_i(t)\, \mathbf{w}_{i\,j} + \mathbf{b}_j\right)\left(1 - \sigma\left(\sum_i \mathbf{o}_i(t)\, \mathbf{w}_{i\,j} + \mathbf{b}_j\right)\right). \tag{2.7}$$

The last term in equation 2.5 is

$$\frac{\partial(\sum_i \mathbf{o}_i(t)\, \mathbf{w}_{i\,j} + \mathbf{b}_j)}{\partial \mathbf{w}_{k\,j}} = \frac{\partial \mathbf{w}_{k\,j}\, \mathbf{o}_k}{\partial \mathbf{w}_{k\,j}} = \mathbf{o}_k. \tag{2.8}$$

The last term in equation 2.6 is

$$\frac{\partial(\sum_i \mathbf{o}_i(t)\, \mathbf{w}_{i\,j} + \mathbf{b}_j)}{\partial \mathbf{b}_j} = 1. \tag{2.9}$$

The gradient of cost function with respect to o_j is

$$\frac{\partial C}{\partial \mathbf{o}_j} = \sum_{l \in L}\left(\frac{\partial E}{\partial \left(\sum_i \mathbf{o}_i(t)\, \mathbf{w}_{i\,l} + \mathbf{b}_l\right)} \frac{\partial \left(\sum_i \mathbf{o}_i(t)\, \mathbf{w}_{i\,l} + \mathbf{b}_l\right)}{\partial \mathbf{o}_j}\right) = \sum_{l \in L}\left(\frac{\partial E}{\partial \mathbf{o}_l} \frac{\partial \mathbf{o}_l}{\partial(\sum_i \mathbf{o}_i(t)\, \mathbf{w}_{i\,l} + \mathbf{b}_l)}\right), \tag{2.10}$$

where all neurons in L receive input from neuron j. Therefore, the derivative with respect to $\mathbf{o}_j$ can be calculated if all derivatives with respect to the output $\mathbf{o}_l$ of the next layer are known. The gradient of cost function with respect to the output of neuron j in the output layer is

$$\frac{\partial C}{\partial \mathbf{o}_j} = \frac{\partial C}{\partial \mathbf{y}} = \frac{\partial}{\partial \mathbf{y}} \frac{1}{2}(\mathbf{t} - \mathbf{y})^2 = \mathbf{y} - \mathbf{t}. \tag{2.11}$$

This is why it is called backpropagation. Backpropagation distributes the error term back up through the layers by modifying the parameters of networks. Gradient descent with backpropagation makes the training of multi-layer networks feasible and

efficient. However, it is not guaranteed to find the global minimum of the cost function but LeCun et al. (2015) argue that in many practical problems, it is not a problem. In addition to square error function, the most common error function for classification problem is the cross-entropy

$$H(\mathbf{y}, \mathbf{p}) = -\sum_i \mathbf{y}_i \log(\mathbf{p}_i), \tag{2.12}$$

where the true probability $\mathbf{y}_i$ is the true label, and the distribution $\mathbf{p}_i$ is the output from neural network. A variant of stochastic gradient descent, which is used commonly in machine learning is Adaptive Moment Estimation (Adam) (Kingma and Ba, 2014). Adam computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. Adam's advantages are that the magnitudes of parameter updates are invariant to rescaling of the gradient, its stepsizes are approximately bounded by the stepsize hyperparameter, it does not require a stationary objective, it works with sparse gradients, and it performs a form of step size annealing (Kingma and Ba, 2014). Three parameters of the Adam optimizer are learning rate α and exponential decay rates for the moment estimates β_1, β_2 in [0,1). First and second moment vectors m_0, v_0 are initialized with 0. These moment estimates are updated using gradients at time t $\mathbf{g}_t$ of cost function with respect to the parameters θ

$$\begin{aligned} \mathbf{m}_t &= \beta_1 \, \mathbf{m}_{t-1} + (1 - \beta_1) \, \mathbf{g}_t, \\ \mathbf{v}_t &= \beta_2 \, \mathbf{v}_{t-1} + (1 - \beta_2) \, \mathbf{g}_t^2. \end{aligned} \tag{2.13}$$

Then the bias-corrected moment estimates are calculated

$$\begin{aligned} \hat{\mathbf{m}}_t &= \frac{\mathbf{m}_t}{1 - \beta_1^t}, \\ \hat{\mathbf{v}}_t &= \frac{\mathbf{v}_t}{1 - \beta_2^t}. \end{aligned} \tag{2.14}$$

The parameters θ are updated $\theta_t = \theta_{t-1} - \frac{\alpha \hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}$ where ϵ is a small constant. Softmax cross entropy cost function with Adam optimizer are used in my proposed method for automatic channel detection described in next Chapters.

Convolutional Neural Networks

Convolutional Neural Networks (Fukushima, 1980) were popular from the introduction of back-propagation in LeCun et al. (1989) to learn the convolution kernel parameters directly from hand-written-number images. CNNs work best for images, which allow to encode certain properties into the architecture, make the forward function more efficient to implement, and vastly reduce the amount of parameters in the network. The fully-connected structure in regular artificial neural networks does not scale to large images. For example, an image of size $200 \times 200 \times 1$ would lead to neurons that have 200*200 = 40000 weights. It is even worse for 3D inputs. However, to make the implementation more rapidly, we would want to have only several neurons. Therefore, the full connectivity in regular neural networks is wasteful and huge number of parameters can easily lead to overfitting. On the other hand, the layers of a CNN have neurons arranged in three dimensions for images: width, height, number of channels (Figure 2.4); and four dimensions for 3D inputs: width, height, depth, number of channels. Some layers in CNNs have parameters and other don't. CNNs transform the original image or 3D input volume layer by layer from the original pixel values to the target outputs. The neurons in a layer are only connected to a small region of the layer before it, which reduces the number of parameters. Three main types of layers in a CNN are convolutional layer, pooling layer, and fully-connected layer.

The convolutional layer consists of learnable filters that are small spatially but

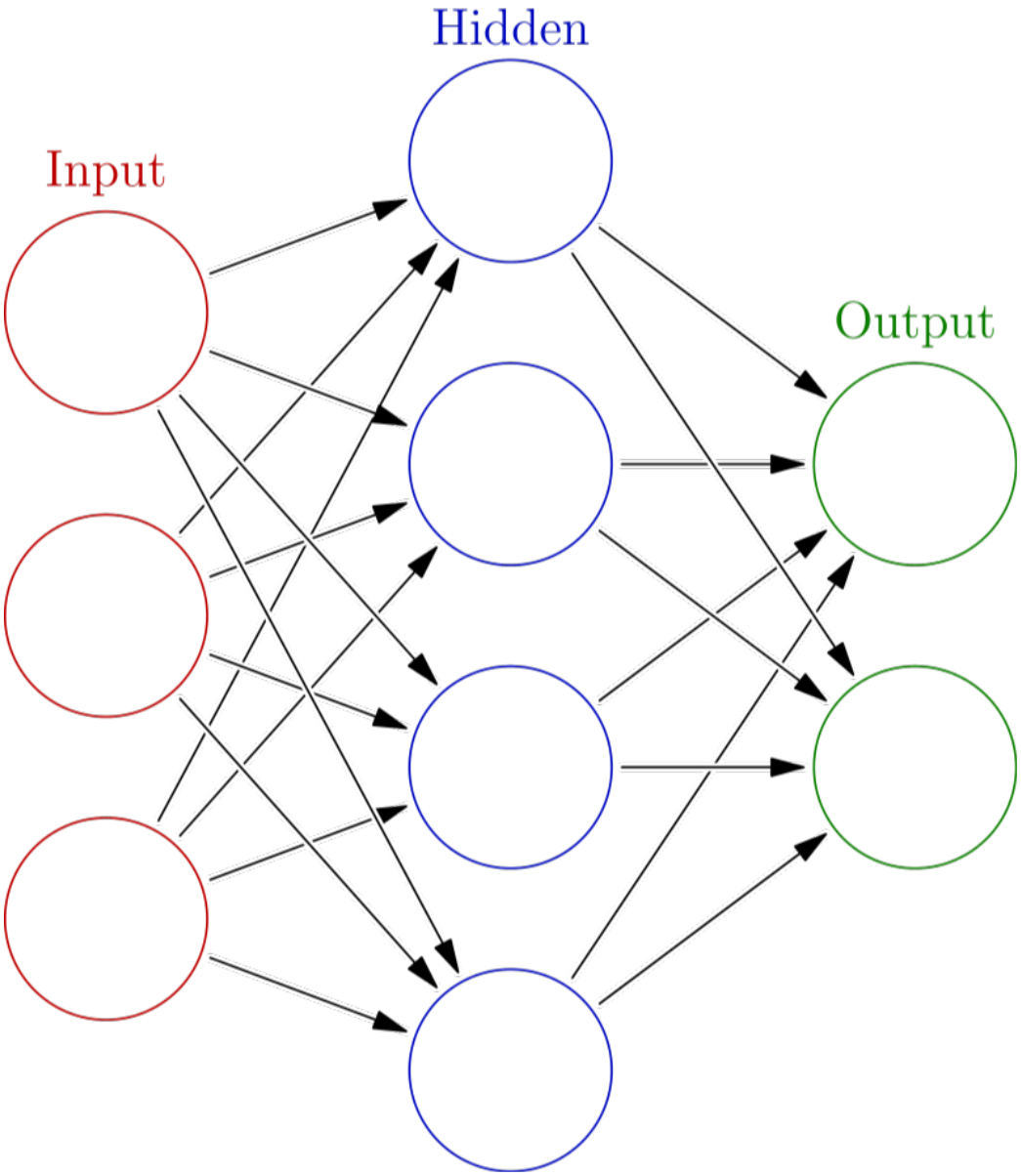


Figure 2.1: An artificial neural network. ch02-MLreview/./cnn ann

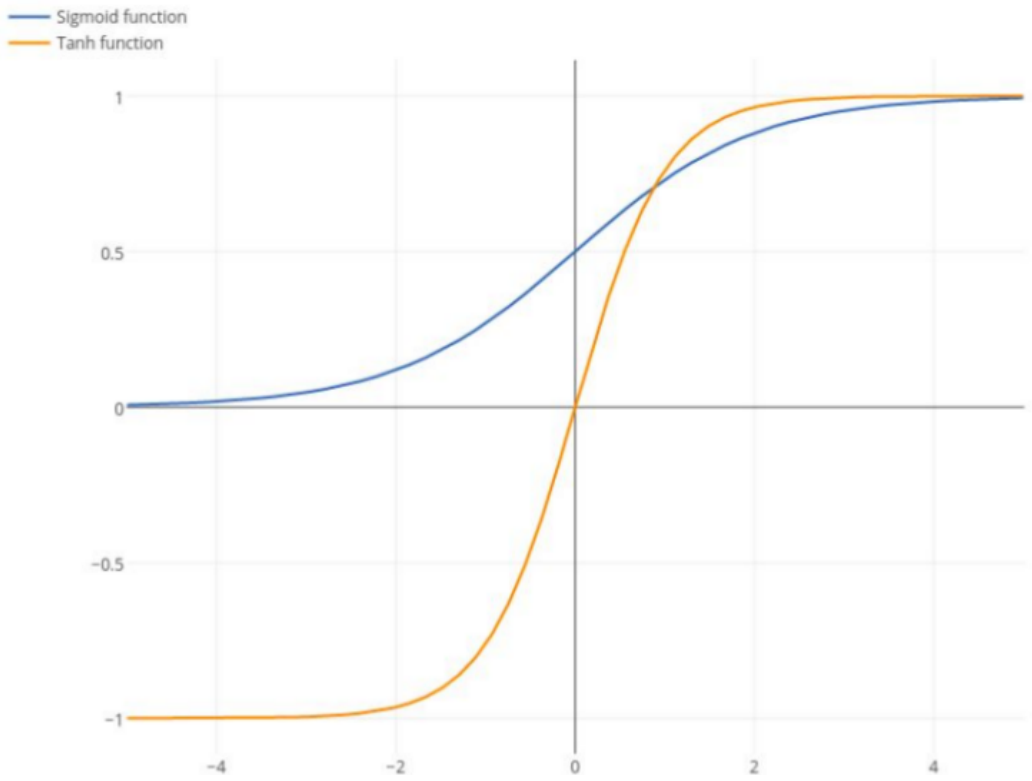


Figure 2.2: Orange: Hyperbolic tangent function. Blue: Sigmoid function. ch02-MLreview/./cnn tanh

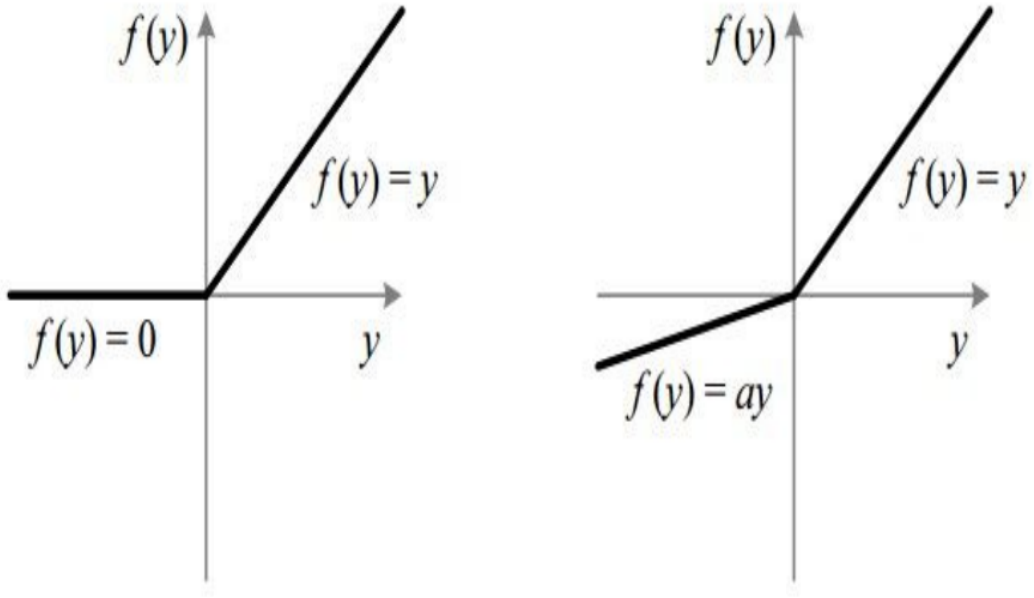


Figure 2.3: Left: Rectified linear unit function. Right: Leaky rectified linear unit function. ch02-MLreview/./cnn relu

extend through the full depth of the input volume. The connections between filters and inputs are local in space but always full along the last dimension of the inputs. Convolutional layers replace matrix multiplication in traditional neural networks with a convolution operator to focus on locality and spatial relationship. Convolution operator in CNNs is not the same as what is used in geophysics and signal processing. It is instead a cross-correlation, in which the filter is slid across the width, height of 2D inputs and the width, height, depth of 3D inputs to compute dot products between its entries and the input at all positions. The results are activation maps that give the responses of that filter at every spatial location. The filter is activated when it sees some types of visual features such as edges of stratigraphic structures in seismic. The output volume from a convolutional layer is a stack of these activation maps along the last dimension. The dimension of output volume depends on three hyperparameters:

the number of feature maps, the stride, and zero-padding. The number of feature maps corresponds to the number of filters. Each filter looks for different things in the input. There is a constraint that the neurons in each feature map share the same weights and biases. Therefore, it dramatically reduces the number of parameters in the network. The stride controls how the convolutional filter is slid. When the stride is one, the filters move one pixel at a time. When the stride is two, the filters jump two pixels at a time and produce smaller outputs spatially. Sometimes it is convenient to pad the input with zeros around the border to control the spatial size of the output. In short, the output dimensions are computed by

$$\mathbf{O} = \frac{\mathbf{W} - \mathbf{F} + 2\,\mathbf{P}}{\mathbf{S}} + 1, \tag{2.15}$$

where $\mathbf{O}$ is the spatial dimensions of outputs, $\mathbf{W}$ is the spatial dimensions of inputs, $\mathbf{F}$ is the spatial dimensions of convolutional filters, $\mathbf{S}$ is the stride, and $\mathbf{P}$ is the amount of zero padding. The backpropagation of a convolution operator is also a convolution but with spatially-flipped filters. It is common to add a non-linear activation function such as ReLU or sigmoid after the convolutional layer. Sigmoid function is differentiable, bounded, and has non-negative derivative at each point. It is suitable for binary classification problem. However, sigmoid function saturates when the input becomes positively or negatively large. ReLU function can solve this problem and prevent networks from vanishing gradient problem. ReLU also has problems as it is not differentiable at zero, unbounded and still suffers from vanishing gradient problem when inputs are negative. However, ReLU is differentiable anywhere else and a value of 0 or 1 can be chosen arbitrarily to fill the point where inputs are 0. LeakyReLU can be used to assign a small positive slope to the negative side (Figure 2.3). It is also common to have a batch normalization layer between a convolutional and a nonlinear activation layer. Batch normalization layer increases the stability of the

neural network by subtracting the batch mean and dividing by the batch standard deviation. The normalized outputs are multiplied by a learned standard deviation parameter γ and add a mean parameter β (Ioffe and Szegedy, 2015). Therefore, the batch normalization layer also has a slight regularization effects to control overfitting.

A pooling layer is periodically inserted in between two successive convolutional layers. It reduces the spatial size of input representation, reduces the amount of parameters, computational cost of the network, and hence controls overfitting. It also creates a representation that is invariant to distortion, translation, and transformation. As the result, the network can easily capture features in the input such as channel structure in seismic volumes despite their locations. Common types of pooling are max-pooling and average-pooling. The common size of a pooling filter is 2x2 (or 2x2x2) with stride of two. The filter captures a max over four numbers in max-pooling layer and an average over four numbers in average-pooling layer. The last dimension of input remains the same. The backward pass for a $\max(x, y)$ operator only routes the gradient to the input that had the highest value in the forward pass. Therefore, during the forward pass of a max-pooling layer, it is common to keep track of the index of maximum activation so that the gradient routing is efficient during backpropagation.

A fully-connected layer has full connections to all activations in the previous layer, similar to regular neural networks. Its output is computed by matrix multiplication followed by a bias offset. The differences between convolutional layer and fully-connected layer are the local connectivity and parameter sharing characteristics in the convolutional layer but the output is still computed by dot products. Therefore, fully-connected layers are convolutional layers with 1x1 filters (or 1x1x1 filters). Softmax activation function can be used in the last layer to map the non-normalized

output to a probability distribution over predicted output classes.

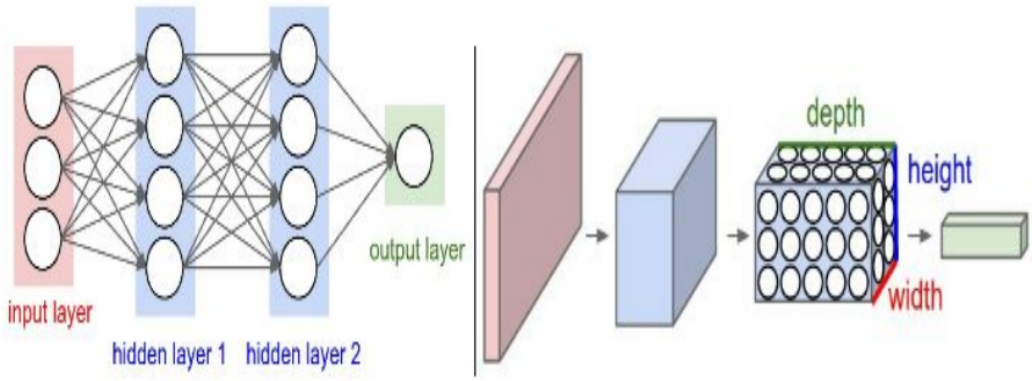


Figure 2.4: Comparison of neuron arrangements. Left: Regular neural networks. Right: Convolutional neural networks. ch02-MLreview/./cnn cnn

Classification and Segmentation

Image classification is the task of assigning an input image one label from a fixed set of categories. The output is the probability of how confident the image belongs to a specific class. Objects of interest are mainly in the middle of input images. A classification model must be invariant to the cross product of these irrelevant variations

a) Viewpoint variation: The object of interest can be oriented in many ways.

b) Scale variation: The object of interest can have different sizes in nature.

c) Deformation: Many objects of interest are not rigid bodies and can be deformed in extreme ways.

d) Occlusion: Sometimes only a small portion of an object is visible.

e) Illumination conditions: The effects of illumination are drastic on the pixel level. For seismic images, this conditions can be thought as the noise in images.

f) Background clutter: The objects of interest may blend into their environment,

making them hard to identify.

g) Intra-class variation: There are many different types of an object of interest, each with their own appearance.

However, the neural network must be sensitive to variations that are important for discrimination (inter-class variations). Instead of trying to specify what every one of the categories of interest looks like directly in the code, the common machine learning approach for image classification problems is a data-driven approach, in which many examples of each class are given to the computer, the learning algorithms look at these examples and learn about the visual appearance of each class. Target output of classification problems is a one-hot vector which has 1 at index corresponding to true label and 0 at other indices.

ML and DL are not only used for whole-image classification, but they are also making progress on local tasks with structured output such as object detection and semantic segmentation. Object detection tasks include advances in bounding box object detection, key-point prediction, and local correspondence (Redmon et al., 2016; Zhang et al., 2014; Long et al., 2014b). The natural progression from coarse bounding box to fine inference is to make prediction at every pixel in the image. Image segmentation understands an image at pixel level, in which each pixel in the image is assigned to a label class. Image segmentation is a combination between semantics and location: global information resolves "what" while local information resolves "where". Image segmentation can be reduced to an image classification problem, where a small sliding window, whose middle is every pixel in the image, goes across the image. However, the target output of segmentation problems is not a one-hot vector. The target has the same size as input size, whose each pixel has its own label class. Automatic channel detection in seismic images can be formularized as an image

segmentation problem in which there are two labels of channel and non-channel.

CONVOLUTIONAL NEURAL NETWORKS FOR SEGMENTATION PROBLEMS

Several CNN architectures have been used for image segmentation in computer vision. In this section, I discuss two common CNNs named fully convolutional networks (Shelhamer et al., 2017) and encoder-decoder convolutional networks (Badrinarayanan et al., 2015; Ronneberger et al., 2015). The latter architecture is the core idea of my method for automatic channel detection in this thesis. Fully connected layers in convolutional neural networks can be viewed as convolutional layers with filters that cover the entire input regions. The computation is highly amortized over the overlapping regions of input patches. Therefore, fully connected layers in neural networks for classification become fully convolutional layers which take input of any size and output classification maps. However, the dimensions of these classification maps are reduced by subsampling using strides or pooling layers. Therefore, for pixelwise prediction, these coarse maps need to be connected back to the pixels. Shelhamer et al. (2017) proposed to use backwards strided convolution for upsampling. Upsampling with factor f is convolution with a fractional input stride of $\frac{1}{f}$. Therefore, backwards convolution with an output stride of f is a way to upsample. Backwards convolution is sometimes called transpose convolution (Dumoulin and Visin, 2016) . It simply reverses the forward and backward passes of convolution. The backwards convolution filters are initialized with bilinear interpolation filters and can be updated during training. However, the outputs are still coarse because of information loss in downsampling so skips combining the final prediction layer with lower layers with finer strides are added to the networks. Combining fine layers and coarse layers lets

the networks make local predictions that respect global structure.

Encoder-decoder convolutional neural networks such as SegNet and U-Net have an encoder and a decoder (Figure 4.18). The encoder has various convolutional layers followed by max-pooling layers to gradually reduce the input dimensions and capture useful features from the input. It does not have the fully convolutional layers so there are lesser parameters compared to fully convolutional networks. The decoder has various upsampling layers followed by convolutional layers. SegNet uses unpooling for upsampling layers in the decoder, in order to use and keep high frequency details intact in the segmentaion. During pooling in the encoder, the indices of maximum values are memorized. During unpooling, the values at these indices are kept intact to produce sparse feature maps. These feature maps are then convolved with a trainable decoder filter bank to produce denser feature maps. Fully convolutional networks do not have these convolutional layers after upsampling layers. The decoder layer corresponding to the first encoder layer produces a K-channel feature map where K is the number of classes. The output is then fed to a trainable softmax classifier to produce the probabilities of each class at every pixel of images. The predicted segmentation corresponds to the class with maximum probability. U-Net, on the other hand, does not reuse pooling indices but instead transfers the entire feature maps to the corresponding decoders for upsampling using transpose convolution. To create denser feature maps, U-Net uses skip connections combining feature maps in the encoder to corresponding upsampled feature maps in the decoder. My method for automatic channel detection in 3D seismic volumes are inspired by SegNet architecture but I expand it for 3D data and use transpose convolution method for upsampling instead of reusing the pooling indices.

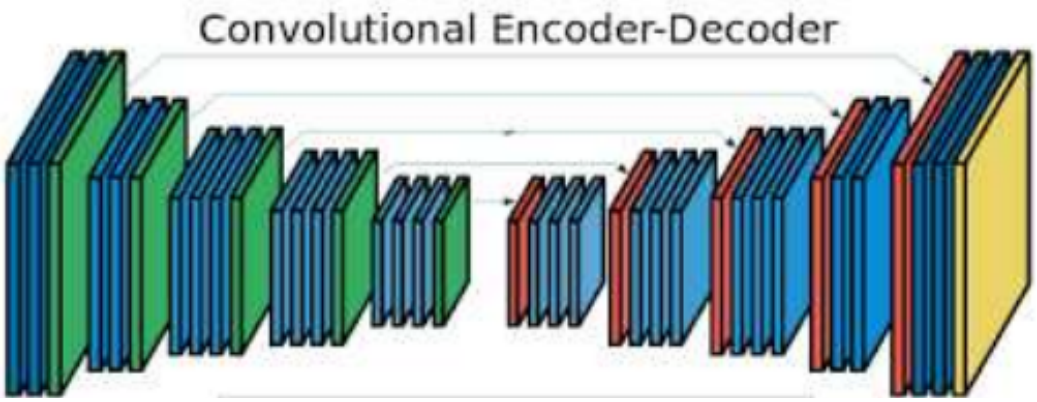


Figure 2.5: Example of an encoder-decoder convolutional neural network. ch02-MLreview/./cnn architecture

BAYESIAN CONVOLUTIONAL NEURAL NETWORKS

Learning in ML and DL is inferring plausible models to explain observed data. A machine can use these models to make predictions with unseen data. Which model is appropriate given the data and the predictions with unobserved data are uncertain. Similarly, making decision based on automatic interpretation results in exploration geophysics is an uncertain process and uncertainty plays an important role. Neural networks can be expressed in a Bayesian way to understand the uncertainty (Ghahramani, 2015). The training phase is the transformation of the prior probability distributions $P(\theta|m)$, defined before observing data, into the posterior distributions $P(\theta|D,m)$, defined after observing data

$$P(\theta|D,m) = \frac{P(D|\theta,m)\,P(\theta|m)}{P(D|m)}, \tag{2.16}$$

where D is the observed data, m is the model, and θ is the network parameters. The prediction can also be expressed by Bayes rule

$$P(x|D,m) = \int P(x|\theta,D,m)\,P(\theta|D,m)\,d\theta, \tag{2.17}$$

where x is new data. Different models can be compared by using Bayes rule

$$P(m|D) = \frac{P(D|m)\,P(m)}{P(D)}. \tag{2.18}$$

The uncertainty of the neural networks comes from all sources such as parameters uncertainty and model structure uncertainty. Bayesian SegNet is a probabilistic image segmentation framework (Kendall et al., 2015). It is a development of SegNet architecture in understanding the network parameters uncertainty by using dropout layers. This is achieved with no additional parameterization. The dropout method randomly removes units in a network (Srivastava et al., 2014), which is a way of getting samples from posterior distribution of softmax class probabilities. Dropout is an approximation of Bayesian inference over the network's weights (Gal and Ghahramani, 2015b). Dropout at training time is a way to prevent overfitting. It can be used at test time to create a Bernoulli distribution over the filter's weights (Gal and Ghahramani, 2015a). It can be considered as Monte Carlo samples obtained from the posterior distribution over models. The mean of this distribution is the segmentation probabilities and the variance of this distribution is the model uncertainties. Monte Carlo sampling produces an overall measure of the model's uncertainty, which is not similar to the relative probabilities between the class labels produced by the softmax function. Kendall et al. (2015) propose to use dropout at the bottleneck between the encoder and decoder. Adding dropout after all the encoder and decoder layers is a too strong regularizer on the model and results in a poorer training fit, test performance. In the lower layers of the networks, basic features are extracted, such as edges and corners, which are consistent across the distribution of models and are better modelled with deterministic weights. The higher level features in the deeper layers, such as shape and textural relationships, are more effectively modelled with Bayesian weights. Bayesian SegNet is used in my method to quantify the uncertainty

of automatic channel detection model in this thesis to provide useful information for interpreters in decision-making.

Machine learning has been used for a long time in geophysics. However, with lots of interpreted data and more powerful computational resources, machine learning and deep learning become very popular in geophysics recently. Many problems in geophysics such as faults, salt, and channel bodies detection can be formulated as classification and segmentation problems in computer vision. Different convolutional neural network architectures have been used for image segmentation. They are mainly different in how common neural network layers are ordered to capture the useful features from the images and upsample extracted features back to the original size of inputs. The method I propose in this thesis for automatic channel detection is inspired by SegNet and Bayesian SegNet architecture.

Chapter 3

Automatic Channel Detection Review

Manual channel detection in 3D seismic volumes is a time-consuming and subjective process. Numerous methods, such as using coherence attribute, sweetness attribute, and instantaneous spectral attribute, have been proposed for helping channel detection in seismic (Wu, 2017; Hart, 2008; Liu and Marfurt, 2007). Channels are distinguished in seismic volumes by lateral discontinuities and by reflectivity contrast with surrounding matrix. In the first section, I discuss some seismic attributes and edge-detection algorithms, such as Sobel filter, for highlighting channel boundaries (Kington, 2015; Bahorich and Farmer, 1995; Marfurt et al., 1998; Gersztenkorn and Marfurt, 1999; Phillips and Fomel, 2017). These methods only focus on channel edges but not channel geobodies. In the second section, I discuss other seismic attributes to highlight channel geobodies but not actually pick them (Hart, 2008; Liu and Marfurt, 2007). All of these attributes are easy to compute but how certain that interpreters can trust on the results from these methods? Uncertainties are important for evaluating the risk of decision-making based on interpretations.

Seismic attributes for enhancing channel boundaries

In this section, I focus on using different algorithms for channel edges enhancement. One of the most popular seismic attributes for highlighting subtle changes in

stratigraphy is seismic coherence. Coherence measures the trace to trace similarity. Similar traces have high coherence coefficients, while discontinuities having low coherence coefficients, such as faults and stratigraphic features, generate sharp delineation along fault planes, reef channel boundaries and deltaic sediments (Chopra, 2002). Different statistical algorithms are used for calculating seismic coherence along some estimate of reflector dips. To demonstrate different coherence algorithm generations, I use a small 3D offshore seismic volume from Parihaka dataset in New Zealand (Figure 3.1). Bahorich and Farmer (1995) use time-lagged cross-correlation to estimate the apprarent dips in inline, crossline direction. The normalized cross-correlations between adjacent seismic traces are combined to estimate the coherence. This algorithm is effective but not as robust as other coherence attributes. Cross-correlation-based coherence enhances channel boundaries in Figure 3.2(a) but the vertical resolution is poor. The second-generation seismic coherence is calculated based on semblance (Marfurt et al., 1998). The semblance is calculated by the ratio of the energy of the average trace within the analysis window to the average energy of the independent traces

$$S[\mathbf{a}] = \frac{\left(\sum_{n=1}^{N} a_n\right)^2}{N \sum_{n=1}^{N} a_n^2}. \tag{3.1}$$

Seismic coherence is defined as the semblance calculated along the dip of the reflector at each analysis point. If the traces are the same across the analysis window, the coherence reaches the maximum value of one. If the traces are random or have polarity changes, the coherence is zero. This method has better vertical resolution because of incorporating a local window of traces (Figure 3.2(b)). However, it is more sensitive to noise such as acquisition footprint and migration artifacts. These correlation-based coherences can be sensitive to lateral amplitude variations.

The third-generation coherence attribute is based on eigenstructure decomposition (Gersztenkorn and Marfurt, 1999). The reflector dips are calculated by more computationally efficient semblance algorithm. The seismic data are flatten in each analysis window and the largest eigenvalue of a covariance matrix between the traces is calculated. Data represented by a single eigenvector have arbitrary amplitude variation but similar waveform within the analysis window. The estimated eigenstructure of coherence is defined as the ratio of energy that can be represented by this single eigenvector to the energy of the independent traces (Chopra, 2002). Therefore, the algorithm is sensitive only to waveform, not amplitude variation, and gives a sharper delineation of discontinuities. It provides better vertical and lateral resolution. However, the method is less efficient as the analysis window becomes smaller (Chopra, 2002). A similar decomposition can be applied to the structure tensor which measures the local covariance of the image in each dimension (Randen et al., 2000). Wu (2017) modifies the traditional gradient-structure-tensor coherence by using directional gradient oriented along seismic structures. Gradient-structure-tensor coherence (Figure 3.2(d)) and eigendecomposition coherence (Figure 3.2(c)) incorporate information about local structures into the calculation but they are significantly computational expensive.

Sobel filter is used in image analysis to efficiently enhance discontinuities and detect edges (Sobel and Feldman, 1968). Sobel filter is a small and integer-valued 3x3 filter (Equation 3.2) which computes an approximation of the gradient of the image intensity function at each point by convolving the data with a zero-phase discrete differential operator and a perpendicular triangular smoothing filter. This filter is used for edge detection in seismic images by orienting the filter along local slopes estimated by maximizing local cross-correlation and dynamically changing the size of

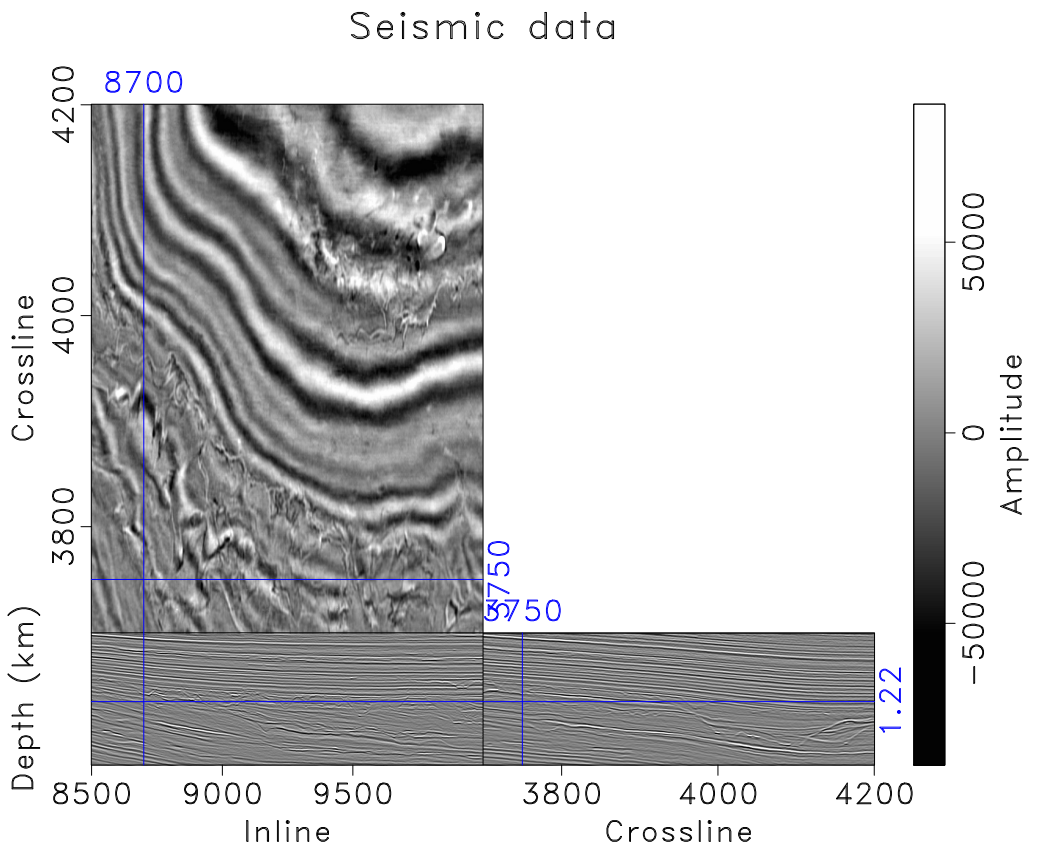


Figure 3.1: 3D seismic field example from Parihaka, New Zealand. ch03-channelreview/./attributes mapped-dn

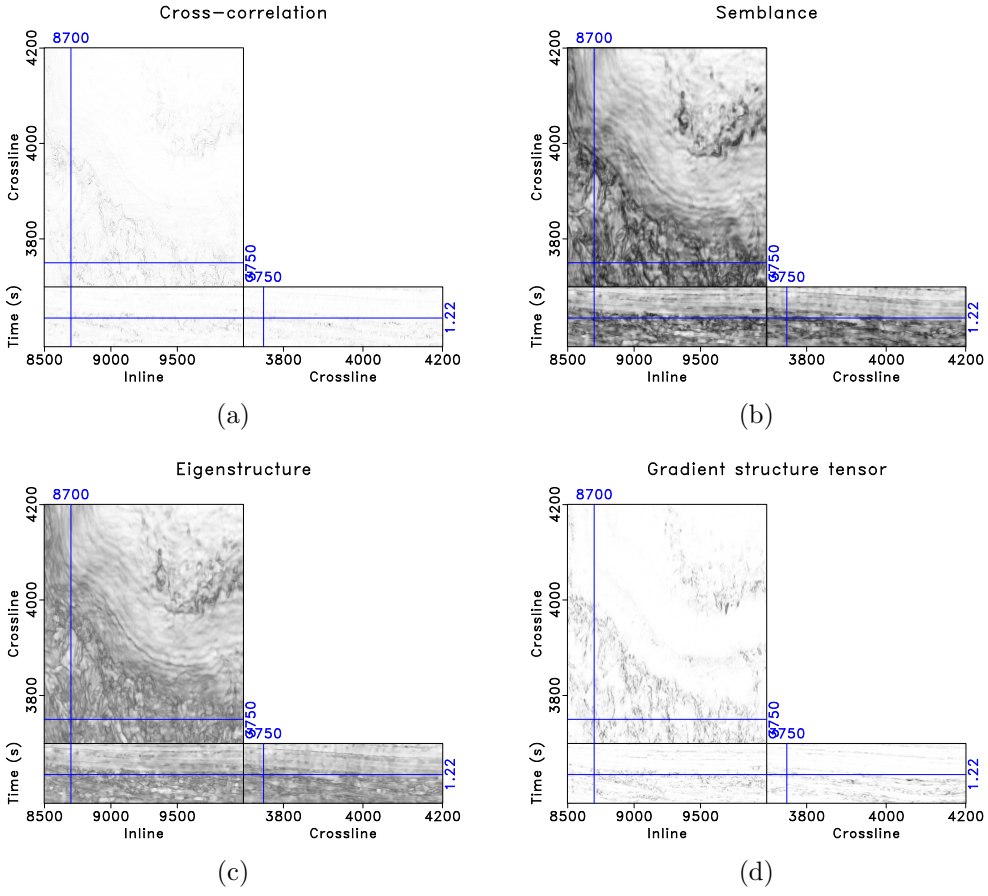


Figure 3.2: Comparison of channel edges enhancement attributes: (a) cross-correlation, (b) semblance, (c) eigenstructure, and (d) gradient-structure tensor. ch03-channelreview/./attributes coh0,coh1,coh2,coh

the filter based on local frequency content (Figure 3.3(a)) (Luo et al., 1996; Aqrawi et al., 2011; Aqrawi, 2014; Aqrawi and Boe, 2011).

$$S_i = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}, \tag{3.2}$$

$$S_x = S_i^{\mathrm{T}} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \tag{3.3}$$

where S_i differentiates in the inline direction and averages in the crossline direction, S_x differentiates in the crossline direction and averages in the inline direction. Phillips and Fomel (2017) modify the Sobel filter by using plane-wave destruction (Fomel, 2002) for replacing discrete differential operator, estimating local slopes (Chen et al., 2013), and using plane-wave shaping (Phillips et al., 2016; Fomel, 2007; Swindeman, 2015) for replacing triangular smoothing. This modification orients the plane-wave Sobel filter along seismic reflection structures. Dip-oriented Sobel filters can be applied directly to a seismic image, cascaded or applied to coherence images to sharpen enhanced edges (Figure 3.3(b) and Figure 3.3(c)) (Chopra et al., 2014). All of these edge-sensitive algorithms are effective for enhancing channel boundaries. However, they do not focus on detecting 3D channel geobodies in seismic volumes and do not have a way to quantify the uncertainty of the results.

Seismic attributes for highlighting 3D channel geobodies

Interpreters are often interested in mapping high-reflectivity channels enclosed in a lower-reflectivity matrix. Therefore, spectral decomposition is used to highlight channels in seismic images because it is sensitive to channel thickness rather than to lateral changes in seismic waveform or amplitude. Shale-dominated intervals tend to have low amplitude with small acoustic impedance contrasts and relatively closely

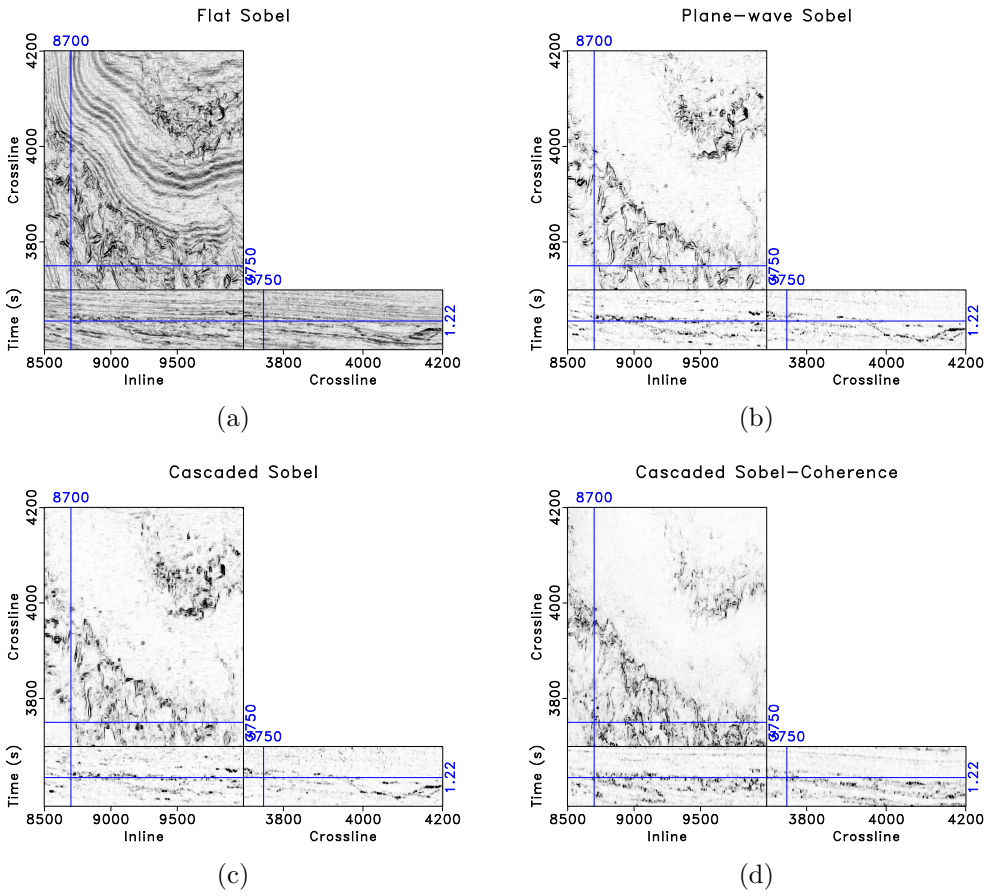


Figure 3.3: Comparison of (a) the traditional Sobel filter, (b) proposed plane-wave Sobel filter, (c) cascaded plane-wave Sobel filter, (d) cascaded plane-wave Sobel filter and coherence. ch03-channelreview/./attributes flat,sobel,sobel2,sobel3

spaced reflections with high frequency. On the other hand, isolated sand channel bodies generate stronger and broader reflections than surrounding shales. The peak frequency slightly increases as the layer thickness decreases. Therfore, to distinguish sands from shales when the contrast in acoustic impedance between those lithologies is high, sweetness attribute defined as the ratio of instantaneous amplitude and instantaneous frequency is used. However, sweetness is less useful for channel detection when the acoustic impedance contrasts between sands and shales are small or when sands and shales are highly interbedded (Hart, 2008). Moreover, the channel geobodies are highlighted but they are not actually picked.

Serveral methods have been proposed for seismic data spectral decomposition such as using local time-frequency decomposition (Liu et al., 2011; Liu and Fomel, 2013), S-transform (Stockwell et al., 1996), regularized nonstationary autoregression (Fomel, 2013), and matching pursuit wavelet-based-spectral-decomposition algorithm (Liu and Marfurt, 2007). The key idea of local time-frequency decomposition is to minimize the error between the input signal and all its Fourier components using regularized nonstationary regression (Fomel, 2009)

$$\min_{A_n} \|f(x) - \sum_n A_n(x)\Psi_n(x)\|_2^2 \,, \tag{3.4}$$

where $A_n(x)$ are the Fourier coefficients and $\Psi_n(x) = e^{i(2\pi nx/L)}$. Short-time Fourier transform (STFT) is a method calculating localized frequency information by computing the Fourier transform with a temporally shifted window (Allen, 1977). The S-transform is similar to the STFT, but with a Gaussian-shaped window whose width scales inversely with frequency

$$S(\tau, f) = \int_{-\infty}^{\infty} S(t) \left\{ \frac{|f|}{\sqrt{2\pi}} \exp\left[\frac{-f^2(\tau - t)^2}{2}\right] \exp(-2\pi i f t) \right\} dt. \tag{3.5}$$

Matching pursuit wavelet-based-spectral decomposition algorithm begins by calculating the instantaneous amplitude and frequency for each input trace and then picks a suite of envelope peaks for major seismic events (Liu and Marfurt, 2007, 2005).

I apply regularized nonstationary autoregression to the Parihaka dataset. Fomel (2013) extends Prony's method (Prony, 1795) to smoothly nonstationary analysis for decomposing seismic signal into components by using regularized nonstationary autoregression. I choose a three-point prediction-error filter to decompose the seismic volume into two most significant data components (Figure 3.4). The corresponding instantaneous frequency is Figure 3.5 and the corresponding amplitude is Figure 3.6. Thin channels in the dataset have middle frequency values and high amplitudes, but are mainly apparent in the low-frequency top volumes. I also apply the local time-frequency decomposition and S-transform to get the spectral decomposition at 20 Hz, 30 Hz, 60 Hz, and 110 Hz (Figure 3.7 and Figure 3.8). For the Parihaka dataset, channels are not clearly identified in the frequency attributes. In addition, frequency attributes are only helpful in distinguishing sands versus shales so not only sand channels are visible in frequency maps, but also normal sand layers. Moreover, frequency attributes are only available for seismic data in time domain. If the data is in depth, velocity model is required to convert depth to time. The frequency attributes are insensitive to channel thickness so they are used simultaneously with coherence attributes to understand both channel width and thickness.

Several seismic attributes can be used for enhancing channel boundaries and highlighting channel geobodies. These attributes are based on the discontinuity of seismic channel boundaries, frequency and amplitude characteristics of sand channel geobodies. However, they only aid interpreters in manually detecting channel geobodies but not actually extract geobodies in seismic volumes. The method I propose

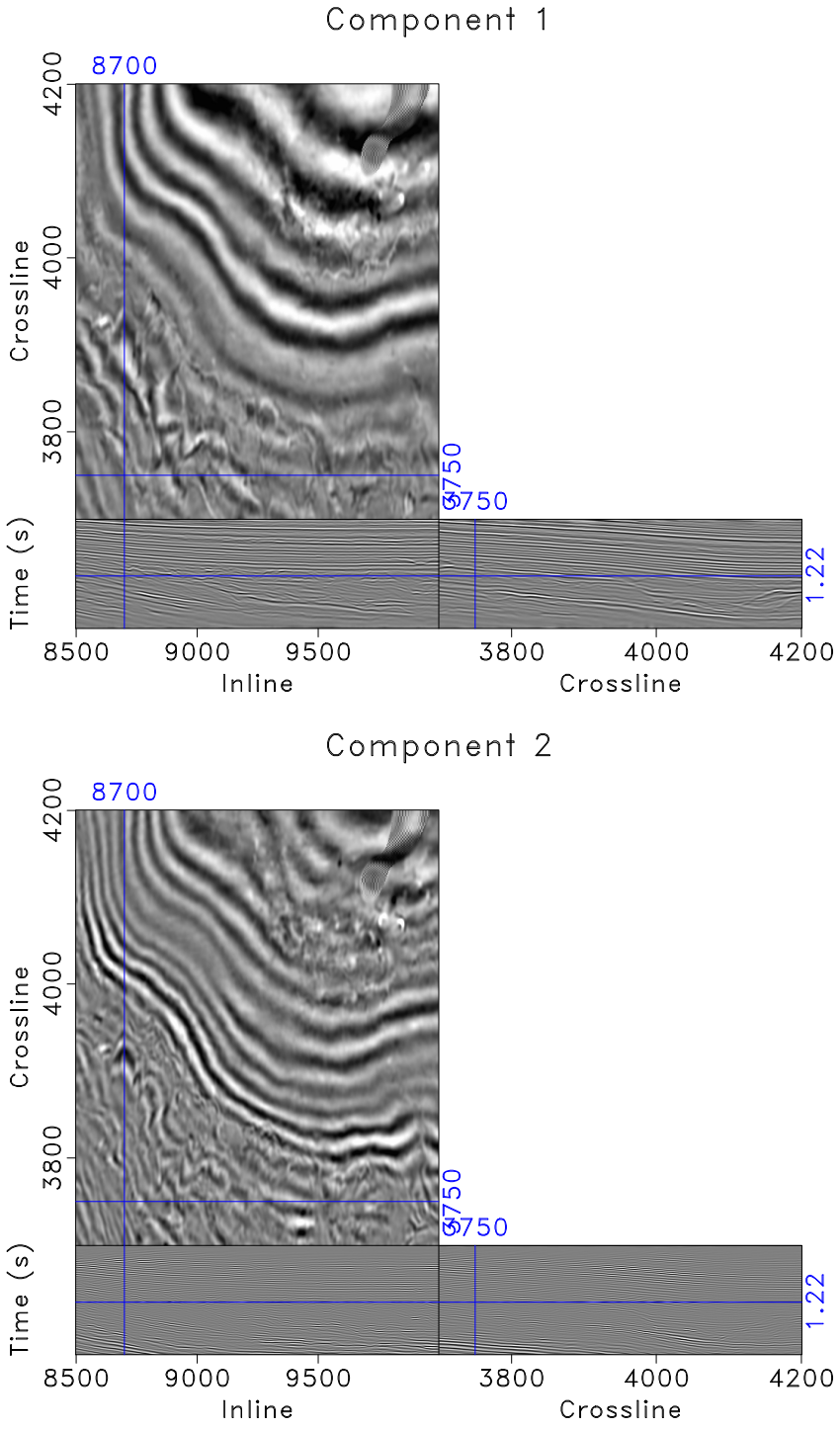


Figure 3.4: Two nonstationary spectral components: high-frequency (Component 2) and low-frequency (Component 1) estimated from the data shown in Figure 3.1. ch03-channelreview/./attributes vsign

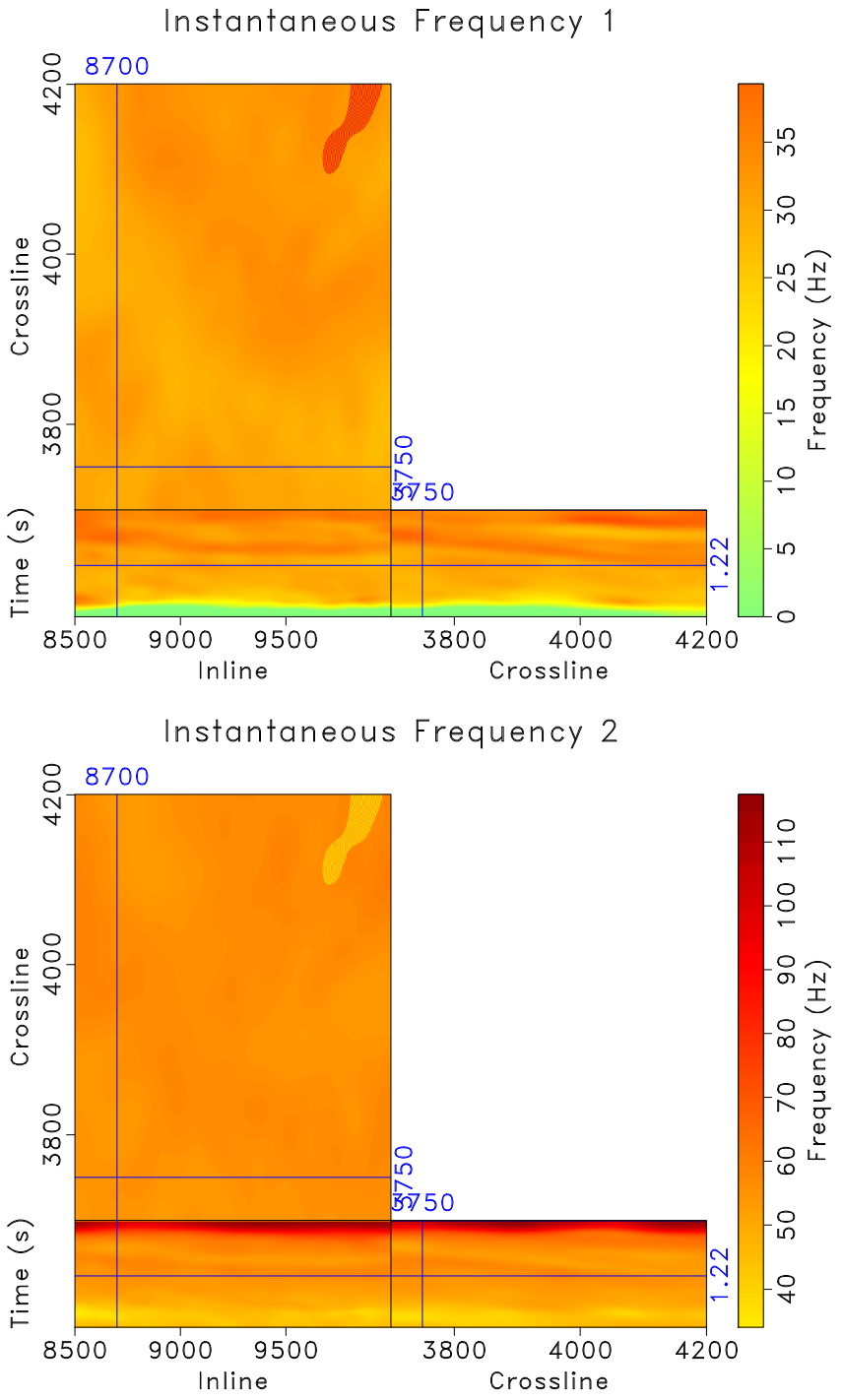


Figure 3.5: Instantaneous frequencies of high-frequency and low-frequency components from decomposition shown in Figure 3.4. ch03-channelreview/./attributes vgroup

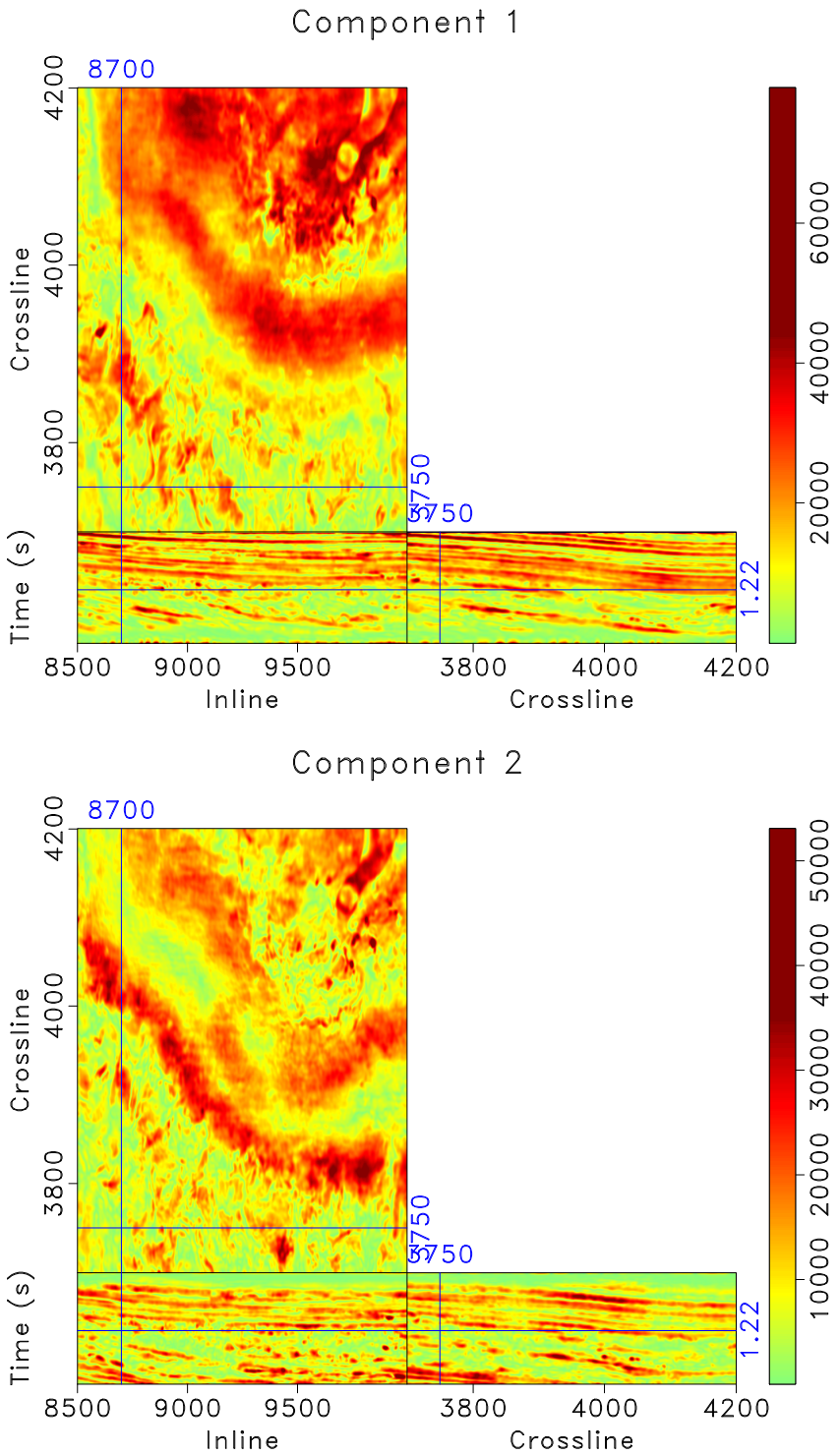


Figure 3.6: Amplitudes of high-frequency and low-frequency components from decomposition shown in Figure 3.4. ch03-channelreview/./attributes vcwht

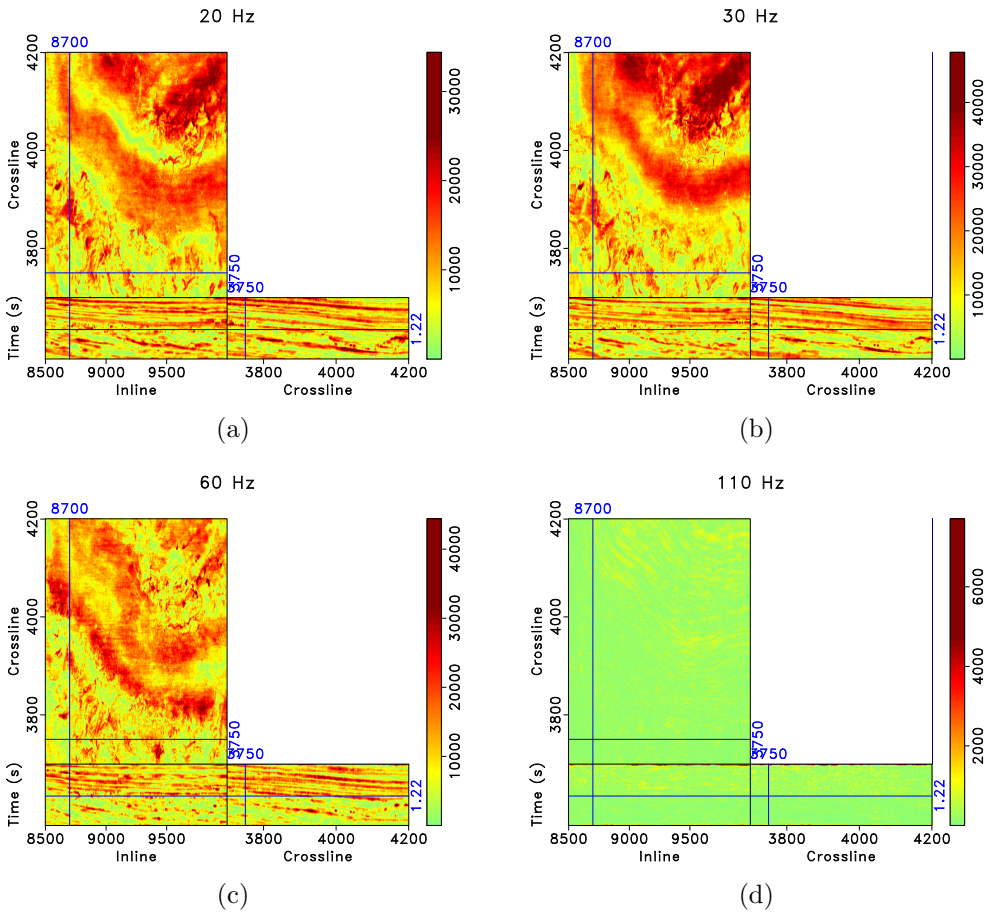


Figure 3.7: The local time-frequency decomposition at (a) 20 Hz, (b) 30Hz, (c) 60 Hz, and (d) 110 Hz. ch03-channelreview/./attributes slice20,slice30,slice60,slice110

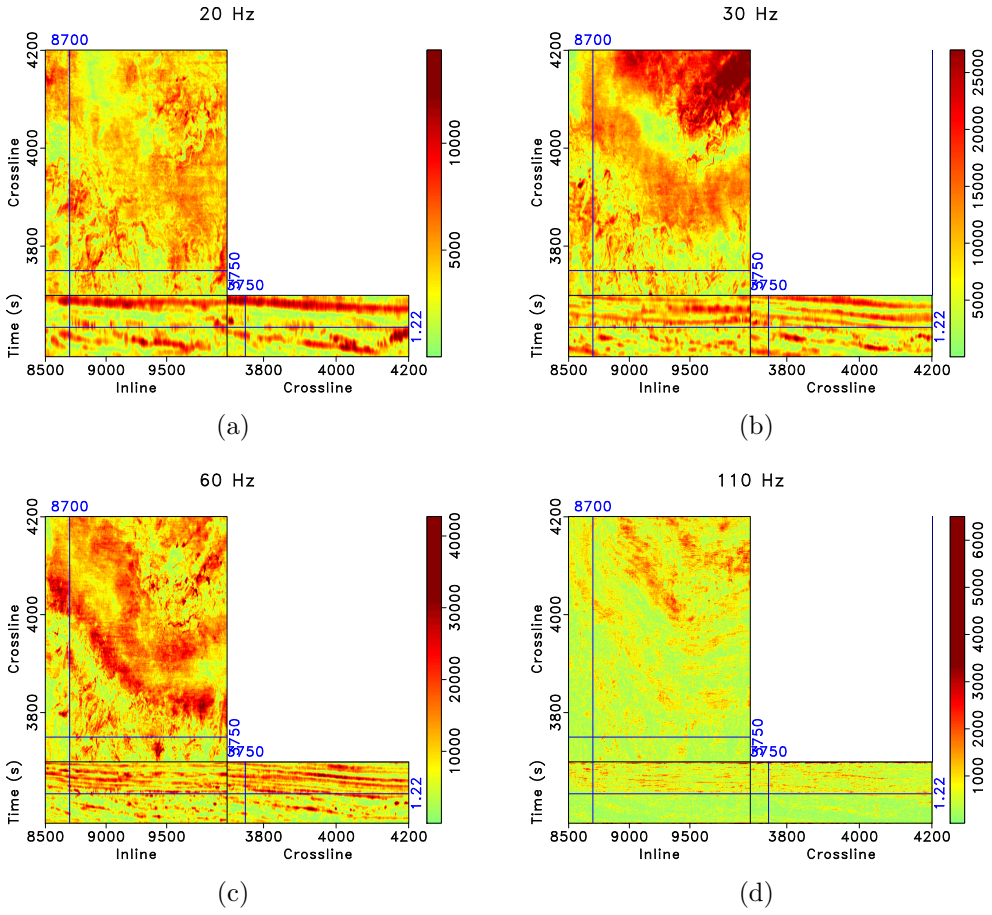


Figure 3.8: The S-transform decomposition at (a) 20 Hz, (b) 30Hz, (c) 60 Hz, and (d) 110 Hz.

ch03-channelreview/./attributes slice-stft20,slice-stft30,slice-stft60,slice-stft110

in this thesis using encoder-decoder convolutional neural network discussed in next chapters can automatically detect channel geobodies at every pixel in the seismic volume with the addition of an uncertainty volume and without any precomputed seismic attributes.

Chapter 4

Automatic channel detection using deep learning - Training phase*

While different seismic attributes described in Chapter 3 are helpful for enhancing channel boundaries and highlighting channel bodies in seismic volumes, these attributes do not actually pick and isolate the 3D channel geobodies. Moreover, it is not easy to quantify the uncertainty of output results from these attributes. Automatic channel detection in 3D seismic volumes is similar to the image segmentation problems in computer vision with two labels of channel and non-channel. Therefore, I propose a deep learning workflow using an encoder-decoder convolutional neural network described in Chapter 2 for automatic channel geobodies detection in 3D seismic volumes. The workflow not only produces the probability volume of channels at every pixel in a 3D seismic volume, but also has the uncertainty volume of how much interpreters can trust the probability results from the neural network. The workflow incorporates knowledge from geologists, geophysicists, and petroleum engineers in creating synthetic data for training. It also reduces the amount of time interpreters use for detecting channel geobodies. Last but not least, it helps interpreters involve in by polishing the results from deep learning networks using the uncertainty volume.

*Parts of this section are published in Pham et al. (2018, 2019). All co-authors contributed equally.

SYNTHETIC DATA FOR TRAINING

Training data

Deep learning is a data-driven technique so data is very important for all deep learning architectures. In computer vision, there are many available benchmark labeled datasets but they are not always available in geophysics. It is best to use interpreted real datasets, but in addition to the fact that they are not always available, manual interpretations of geologists and geophysicists are sometimes biased. Therefore, to have clean training datasets, I propose to create synthetic datasets combining knowledge of geologists, geophysicists, and petroleum engineers. An example of event-based forward model has four architectural stages: valley with maximum erosion and without sediment fill in stage 1, channel amalgation with low rate of aggradation in stage 2, disorganized channel stacking pattern with moderate rate of aggradation in stage 3, and organized channel stacking pattern with high rate of aggradation in stage 4 (McHargue et al., 2010) (Figure 4.1). The training dataset I use in this thesis is a 3D convolutional depth model created by James Jennings in collaboration with Chevron. The data simulates a complex stacked deep-water channel system in Africa.

A group of geologists went to an analog field in California to study the properties distribution of channel system such as channel depth, width of channel top, width of channel bottom, and channel aggradation. Aggradation is the increase in land elevation, typically in a river system, due to the deposition of sediment (Figure 4.2). The channel positions and skews are organized originally in fifteen 2D arrays in `Mathematica` with the grid spans: $0\,m \leq x \leq 5400\,m$ and $0\,m \leq y \leq 3000\,m$. There are total 181 samples in x direction with interval of 30 meters and 101 samples in y direction with interval of 30 meters. The skew and position arrays are interpolated to

a grid of 360 samples with 15 meters interval in x direction and 200 samples with 15 meters interval in y direction. The z coordinate measures height above the bottom of the first channel. The top of the last channel is at 110m. The channel properties are put into layer arrays of the same grid spans. Seismic field data at the depth of channel system in Africa has low resolution so it cannot be used for modelling. Another available field data is the shallow high-resolution seismic data for understanding the channels shape. An analytical curve is used to make channel, bar drape, and margin drape profiles (Figure 4.3(a), Figure 4.3(d), and Figure 4.3(g))

$$\begin{aligned} pr &= \min\left(d\left(zs + \frac{de\left(\tanh(\pi\left(xl - 0.5\right)) + \tanh(\pi\left(xr - 0.5\right))\right)}{2}\right), 0\right), \\ xl &= \frac{2\,p + s\left(wt - wb\right) - wb}{\left(s + 1\right)\left(wt - wb\right)}, \\ xr &= \frac{2\,p + s\left(wt - wb\right) + wb}{\left(s - 1\right)\left(wt - wb\right)}, \end{aligned} \tag{4.1}$$

where d is the channel depth, pr is the profile, zs is the corresponding profile z-shift, de is the corresponding profile depth coefficient, p is the channel position, s is the channel skew, wt is the channel top width, and wb is the channel bottom width. In my model, I set bar drape depth coeffienct to 0.8, bar drape z-shift to -0.15, margin drape depth coefficient to 1.2, and margin drape z-shape to 0.1. Channel surface is calculated by adding channel profile with aggradation (Figure 4.3(b)). Bar drape surface is calculated by adding aggradation with elementwise maximum values between bar drape profile and channel profile (Figure 4.3(e)). Margin drape surface is calculated by adding aggradation with elementwise maximum values between margin drape profile and channel profile (Figure 4.3(h)). These surfaces are then eroded (Figure 4.3(c), Figure 4.3(f), and Figure 4.3(i)).

I then generate 3D layered channel properties models from specified 16 interfaces (an additional interface above the top layer) (Figure 4.4 and Figure 4.5). The z

dimension has 100 samples with 2 meters sampling interval and origin at -45m. The channels topographic relief (Figure 4.5(f)) is generated with addition of channel mask (Figure 4.6(a)), which can be used as training label in next section

$$zrel = \frac{m\,(z-a)}{d}, \tag{4.2}$$

where $zrel$ is the channels topographic relief, m is the above-layer-zero mask, z is the z-dimension of aggradation grid, a is the aggradation grid, and d is the depth grid. Bar drape mask and margin drape mask are generated by subtracting the third dimension from the values of corresponding grids and masking 1 where the subtraction corresponding to channels location (and also not corresponding to bar drape location for margin drape mask) is smaller than 0 (Figure 4.6(b) and Figure 4.6(c)). Position shift and channels height are calculated using position grid, top width grid, and skew grid (Figure 4.7)

$$\begin{aligned} posshift &= \frac{p}{w} + \frac{(asshift)\,s}{2}, \\ asheight &= (asheight0) + ((asheight1) - (asheight0))\,s, \end{aligned} \tag{4.3}$$

where $posshift$ is the position shift, p is the position grid, w is the top width grid, $asshift$ is the lateral shift at channel max bend, $asheight$ is the channels height, $asheight0$ is the height at channel inflection, $asheight1$ is the height at channel max bend, and s is the skew grid. The sand fraction grid is made from these properties models and is padded 40 samples in both directions of x and y axes (Figure 4.8(a))

to have 440 samples in x, 280 samples in y, and 100 samples in z

$$\begin{aligned} sand &= bd\,(bdsand) + md\,(mdsand) + (asmask)\,(assand) \\ &\quad + (namask)\,((nasand0) + ((nasand1) - (nasand0))\,(sandparm)), \\ sandparm &= \frac{-z}{(1 - h + 4^{(asshape)}\,h\,(\frac{x^2}{(aswidth)^2})^{(asshape)}}, \\ asmask &= (sandparm)\,(1 - bd)\,(1 - md), \\ namask &= ch - bd - md - (asmask), \end{aligned} \tag{4.4}$$

where $sand$ is the sand fraction, bd is the bar drape mask, md is the margin drape mask, $bdsand$ is the bypass drape sand fraction, $mdsand$ is the margin drape sand fraction, $assand$ is the amalgamated sand fraction, $nasand0$ is the non-amalgamated sand fraction at channel top, $nasand1$ is the non-amalgamated sand fraction at amalgamated sand contact, z is the topographic relief, h is the channels height, $asshape$ is the cross-section shape factor, x is the position shift, $aswidth$ is the width at channel bottom, and ch is the channels mask. The channel geometrical parameters I used in modeling are: 0.8 for amalgamated sand width at channel bottom and lateral shift at channel max bend, 0.5 for almagamated height at channel inflection, 0.8 for amalgamated sand height at channel max bend, 1.2 for amalgamated sand cross-section shape parameter. The sand fraction parameters are: 0.3 for sand fraction in the bypass drape, 0.2 for sand fraction in the margin drape, 1.0 for sand fraction in the amalgamated sand, 0.4 for sand fraction in the non-amalgamated sand at channel top, and 0.9 for sand fraction in non-amalgamated sand at amalgamated sand surface. Shale fraction and porosity is calculated from sand fraction (Figure 4.8(b) and Figure 4.8(c))

$$\begin{aligned} shale &= 1 - sand, \\ \phi &= 0.378 - 0.144 \times shale - 0.0000262\,(10000 - 7000). \end{aligned} \tag{4.5}$$

Correlated noise is added to the porosity in the channel interval and background (Figure 4.8(d)). Background noise parameters are: 0.03 for porosity noise standard deviation, 1 for covariance shape parameter, [1,0,0], [0,1,0], [0,0,1] for three covariance range orientation vectors, and [1000,1000,1] for covariance range parameters. Sand noise parameters are: 0.01 for porosity noise standard deviation, 1 for covariance taper switch, [1,0,0], [0,1,0], [0,0,1] for three covariance range orientation vectors, and [200,200,1] for covariance range parameters. These parameters together with geostatistics method are used to generate correlated Gaussian random noise with an isotropic stable covariance which is added to the background and channel porosity. Each covariance range parameter controls the correlation range in the direction of the corresponding orientation vector. The method starts with computing the scaled distance

$$\begin{aligned} h &= \sqrt{\left(\frac{u}{ru}\right)^2 + \left(\frac{v}{rv}\right)^2 + \left(\frac{w}{rw}\right)^2}, \\ u &= x \times uniu[0] + y \times uniu[1] + z \times uniu[2], \\ v &= x \times univ[0] + y \times univ[1] + z \times univ[2], \\ w &= x \times uniw[0] + y \times uniw[1] + z \times uniw[2], \end{aligned} \tag{4.6}$$

where ru, rv, rw are covariance range parameters; $uniu$, $univ$, $uniw$ are unit vectors in uvw coordinate system; and x, y, z are normal grid coordinates. Then a stable covariance model for unit variance is made based on calculated distances (Dimitrakopoulos and Luo, 1994)

$$cov(h) = e^{-\left(\frac{h}{r}\right)^{\alpha}}, \tag{4.7}$$

where α is the covariance model shape parameter, h is a lag in a certain direction, and r is the corresponding range parameter. The corresponding stable semi-variogram model is

$$\gamma(h) = 1 - e^{-\left(\frac{h}{r}\right)^{\alpha}}. \tag{4.8}$$

The shape parameter I use is 1, which corresponds to the exponential model of geostatistics. The model produces spatially correlated and continous random fields with intermediate roughness and first derivatives that are discontinuous everywhere. The generated Gaussian random fields have a mean of zero and a variance of one. A cosine taper is smoothly applied to eliminate any non-zero derivatives on the edges of the grid. After applying taper, the coordinate system is rotated back to the original grid origin. The Fourier transforms of the covariance and uncorrelated Gaussian random noise with unit variance created from covariance model are computed. The amplitude spectrum is the square root of the power spectrum. The final step is to combine the amplitude spectrum and noise by performing a weighted moving average of the uncorrelated Gaussian noise, which produces a correlated Gaussian noise. The noise is inverted back using inverse Fourier transform. There is one realization for the background noise (Figure 4.9(a)), and fifteen different realizations for the channels, one for each of the fifteen channels in the model. The fifteen sand realizations are all made with the same set of parameters but with different random number seeds. They are concatenated to have a correlated Gaussian random field for sand (Figure 4.9(b)). After adding noise to the reservoir porosity, densities and sonic velocities are calculated

$$\begin{aligned} \rho &= 2.70 - 0.085 \times shale - 1.75\,\phi, \\ V_P &= 4090 - 960 \times shale - 4510\,\phi, \\ V_S &= -880 + 0.782 \times V_P. \end{aligned} \tag{4.9}$$

Top and bottom tapers are applied to the property arrays to get P-wave velocity and density (Figure 4.10). The properties are linearly interpolated between those generated by the channel model, and a set of constants for the top (or bottom) of the grid. The weighting is computed to produce the constant values at the top (or bottom) of the taper zone, the channel model output at the bottom (or top) of the

taper zone and below (or above), and linear weighting with depth in between. I calculate the acoustic impedance in depth and convert to time using the velocity model (Figure 4.11). Acousic impedance is converted to reflectivity (Figure 4.12) and the reflectivity is convolved with a 40 Hz Ricker wavelet and changed back to depth using velocity model to create the synthetic data (Figure 4.13). The data has 440 samples with 15 meters sampling interval in x direction, 280 samples with 15 meters sampling interval in y direction, and 100 samples with 2 meters sampling interval in z direction.

Training label

To get the location of channels, I eliminate the noise inside the channels (Figure 4.15(b)) and subtract the data without noise from the data with noise (Figure 4.16(a)). To create the training label, I simply mask one where there are channels and zero everywhere else (Figure 4.16(b)). I modify different channels' properties as described above, such as amalgamated sand cross-section shape parameter, porosity, dominant frequency, and channels thickness to create a diverse training dataset (Figure 4.14). Because of limited computational resources, a training batch has 4 seismic volumes. The size of each volume is $156 \times 156 \times 100$ samples (Figure 4.17). Examples in the training data overlap with one another, but it is a way of augmenting the data. I generate a total of 1025 training examples with 6 examples for validating the network. The synthetic data can be more complicated by adding an overburden with stochastically generated velocity fluctuations and correlated noise in porosity and doing exploding reflector modelling (Janson and Fomel, 2011; Fomel et al., 2007). The synthetic data is a combination of different inputs from field geologists, geophysicists, and petroleum engineers. It will become a trend in future interpretation workflow

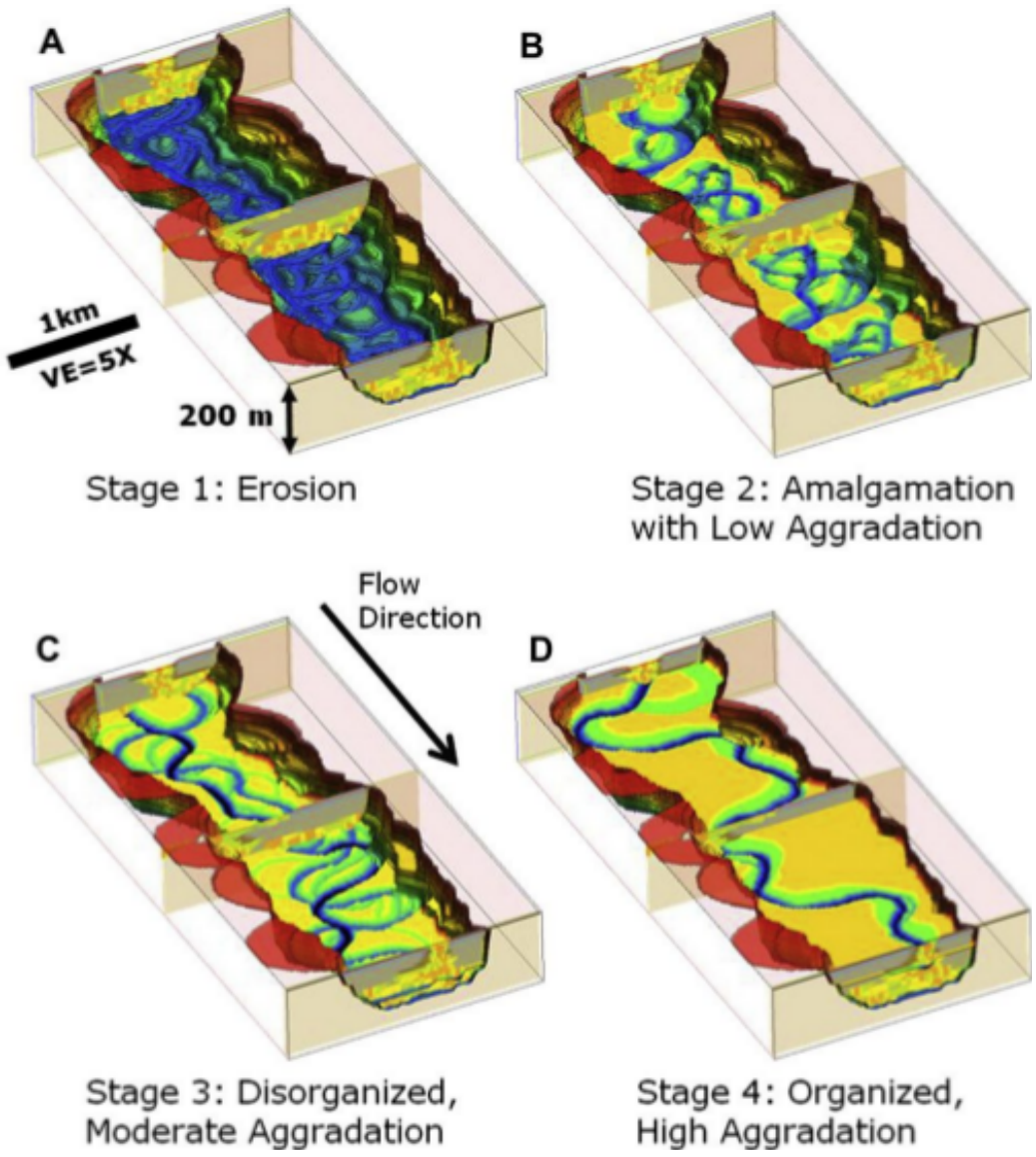


Figure 4.1: Example event-based forward model showing 4 architectural stages. (Figure from McHargue et al. (2010)) ch04-training/./model EB

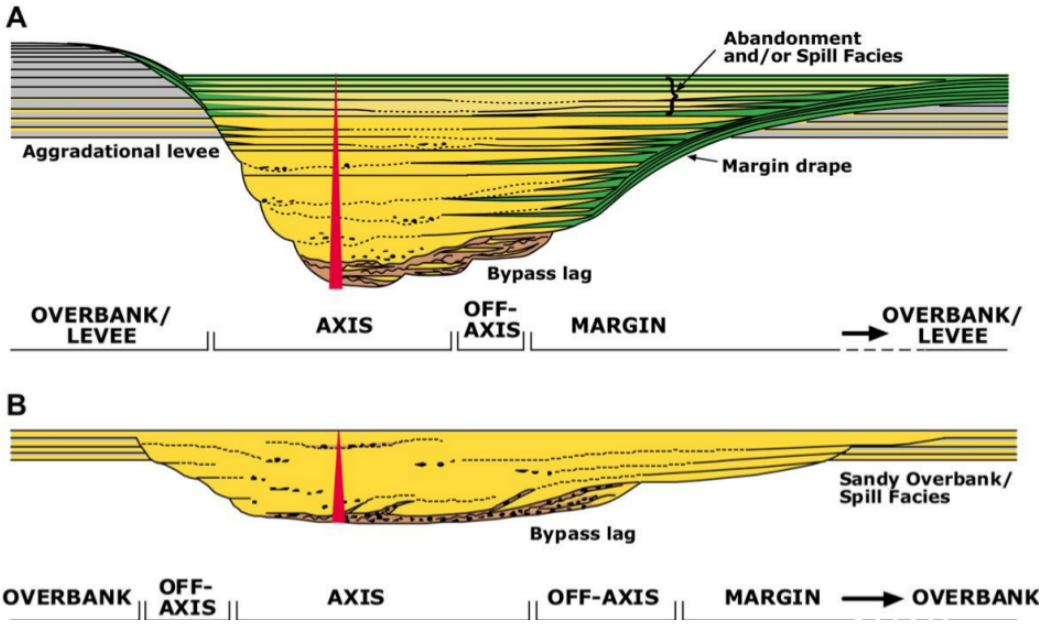


Figure 4.2: Schematic representations of common fill styles of under-filled (A) and filled channel elements (B). (Figure from McHargue et al. (2010)) ch04-training/./model schematic

that neural networks are trained with synthetic data created from knowledge of experts and the trained models are used for doing interpretation with new data such as fault, salt, and channel geobodies detection.

ENCODER-DECODER CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE

My proposed architecture inspired by the SegNet and Bayesian SegNet architecture described in Chapter 2 for automatic channel detection consists of four layers in the encoder and corresponding four layers in the decoder (Figure 4.18). Each encoder layer has a convolutional layer and a pooling layer to learn useful features (Figure 4.19). Convolutional layers have multiple filters that are trainable in the learning

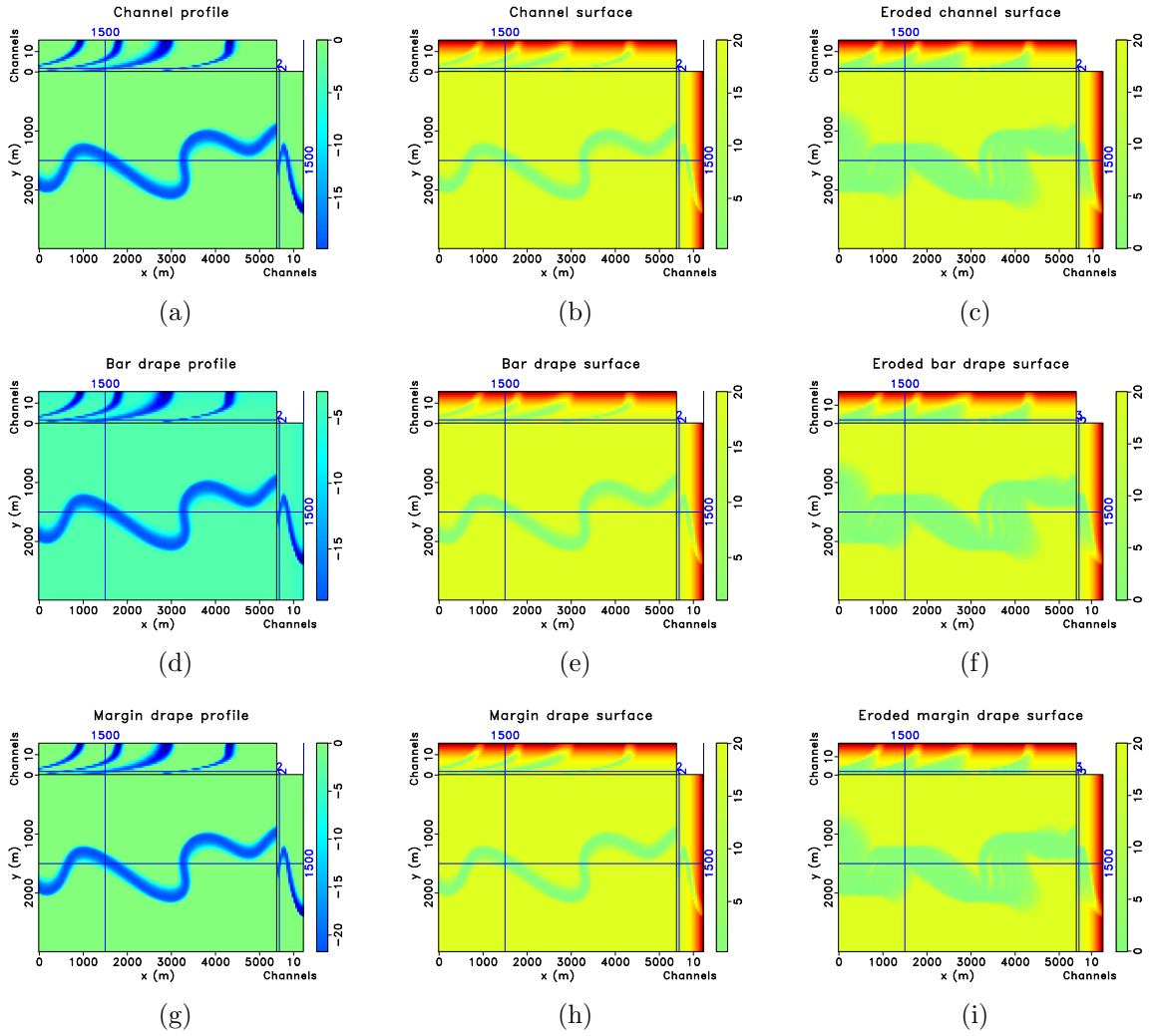


Figure 4.3: (a) Channel profiles. (b) Channel surfaces. (c) Eroded channel surfaces. (d) Bar drape profiles. (e) Bar drape surfaces. (f) Eroded bar drape surfaces. (g) Margin drape profiles. (h) Margin drape surfaces. (i) Eroded margin drape surfaces.

ch04-training/./model ch-profiles,ch-surfaces,ch-erode,bd-profiles,bd-surfaces,bd-erode,md-profiles,md

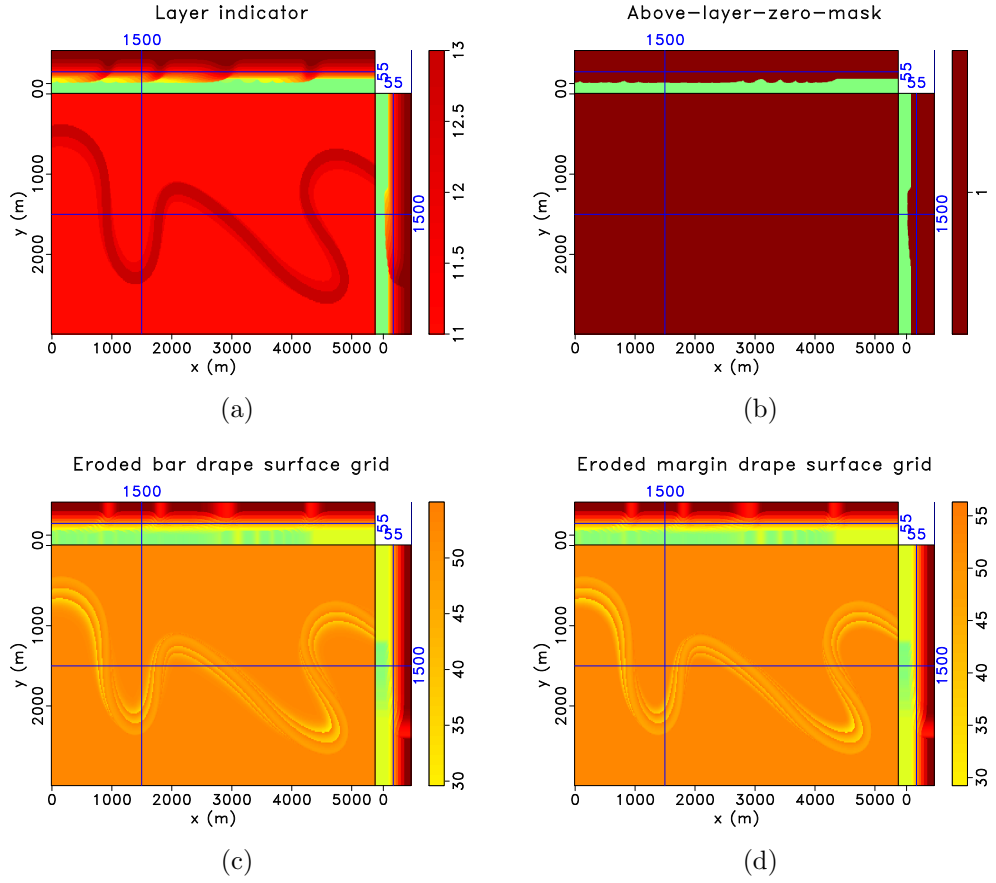


Figure 4.4: (a) Layer indicator. (b) Above-layer-zero mask. (c) Eroded bar drape grid. (d) Eroded margin drape grid. ch04-training/./model layer,layer-00-mask,bd-erode-grid,md-erode-grid

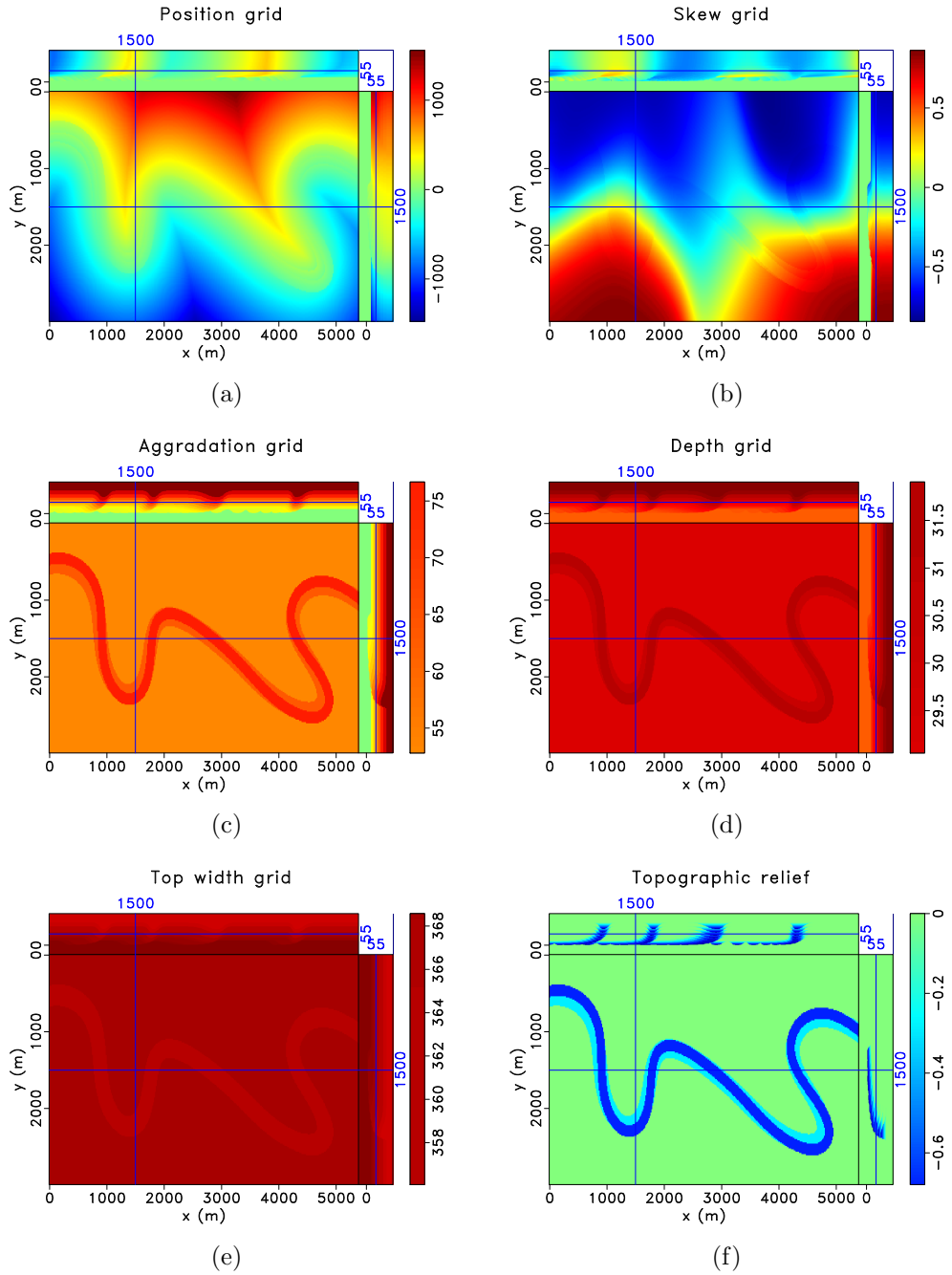
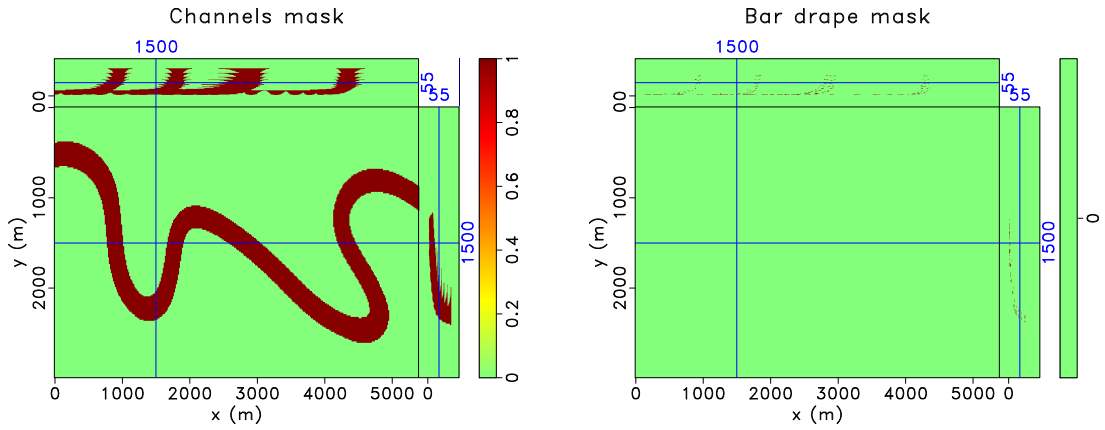
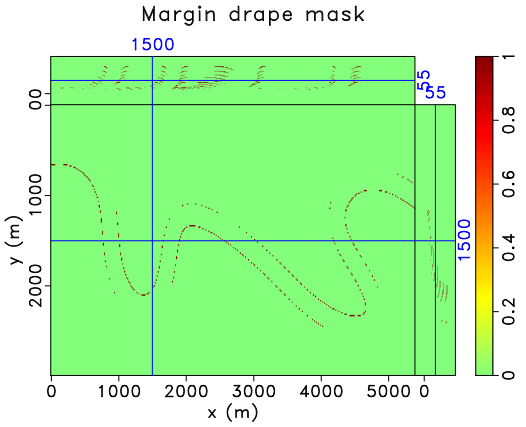


Figure 4.5: (a) Position grid. (b) Skew grid. (c) Aggradation grid. (d) Depth grid. (e) Top width grid. (f) Channels topographic relief. ch04-training/./model pos-grid,skew-grid,aggrade-grid,depth-grid,wtop-grid,z-rel



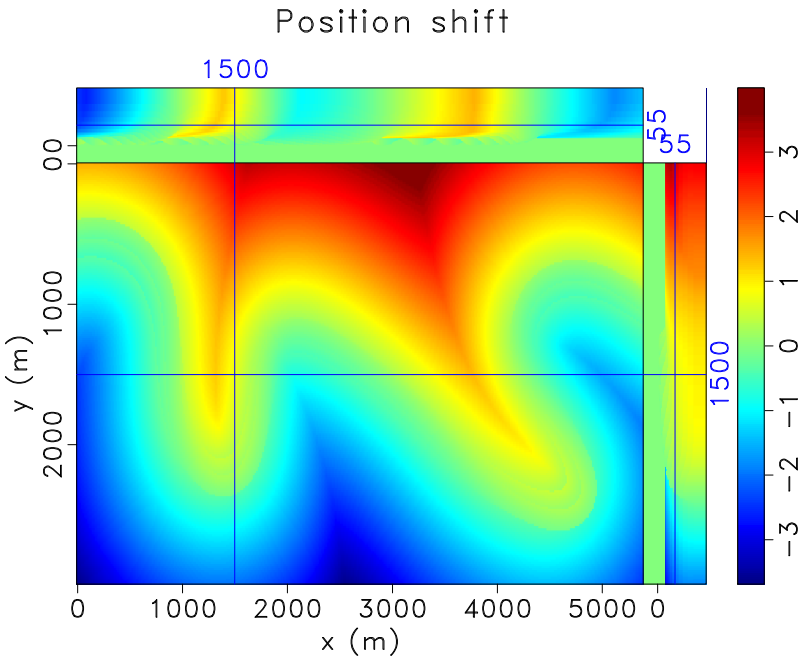
(a)

(b)

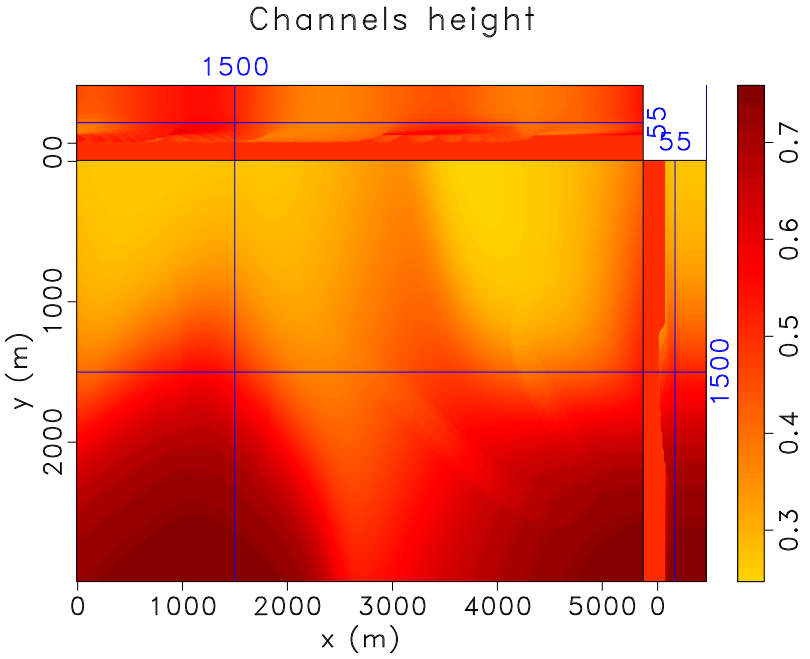


(c)

Figure 4.6: (a) Channels mask. (b) Bar drape mask. (c) Margin drape mask. ch04-training/./model ch-mask,bd-mask,md-mask



(a)



(b)

Figure 4.7: (a) Position shift. (b) Channels height. ch04-training/./model pos-shift,as-height

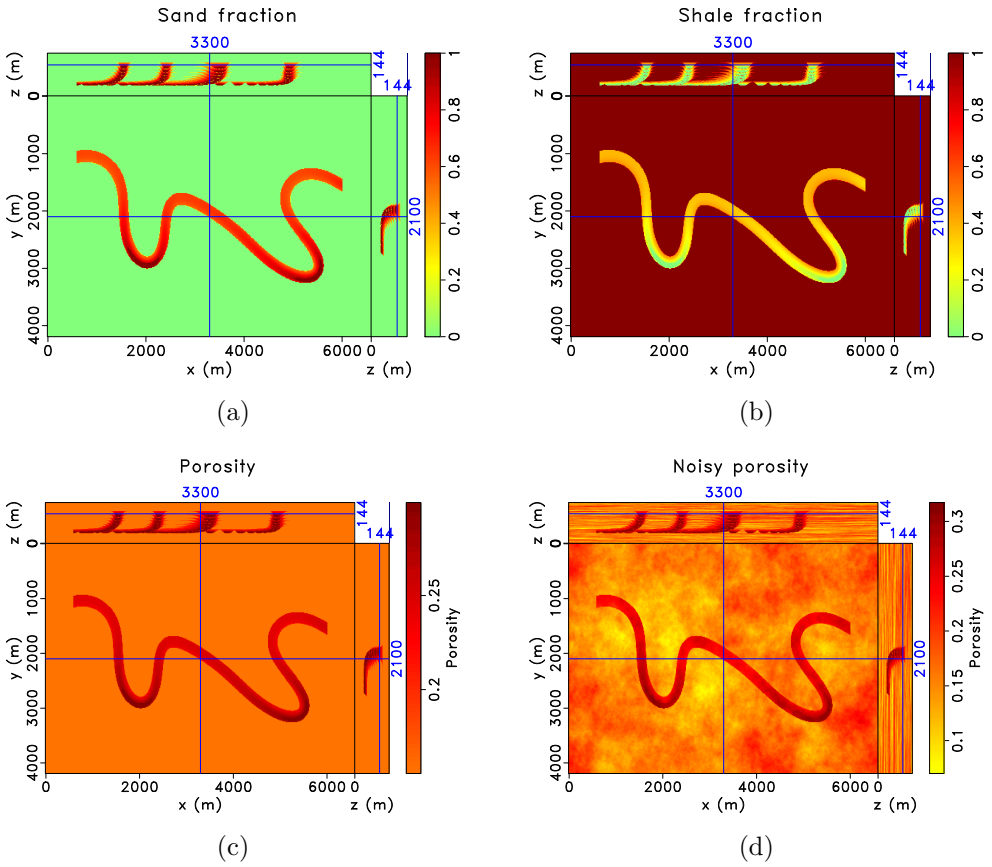
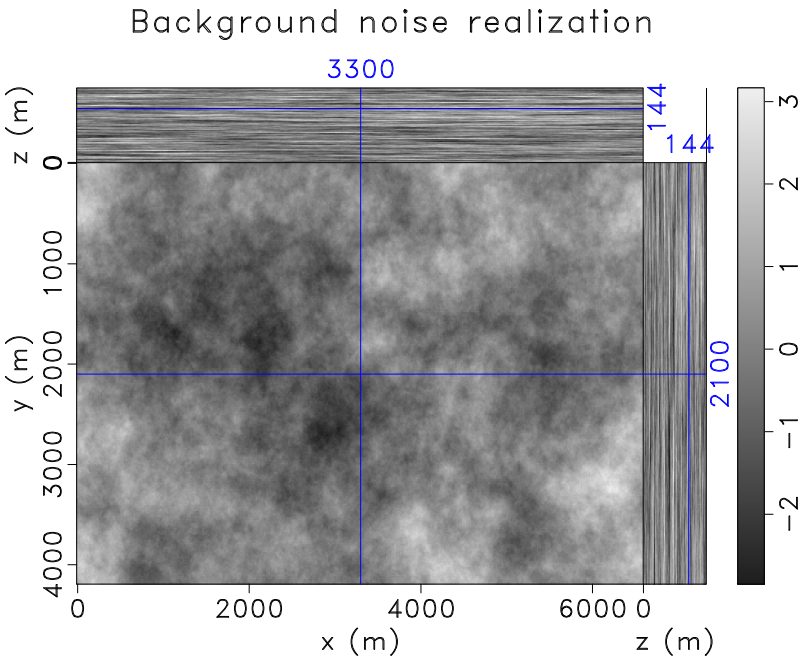
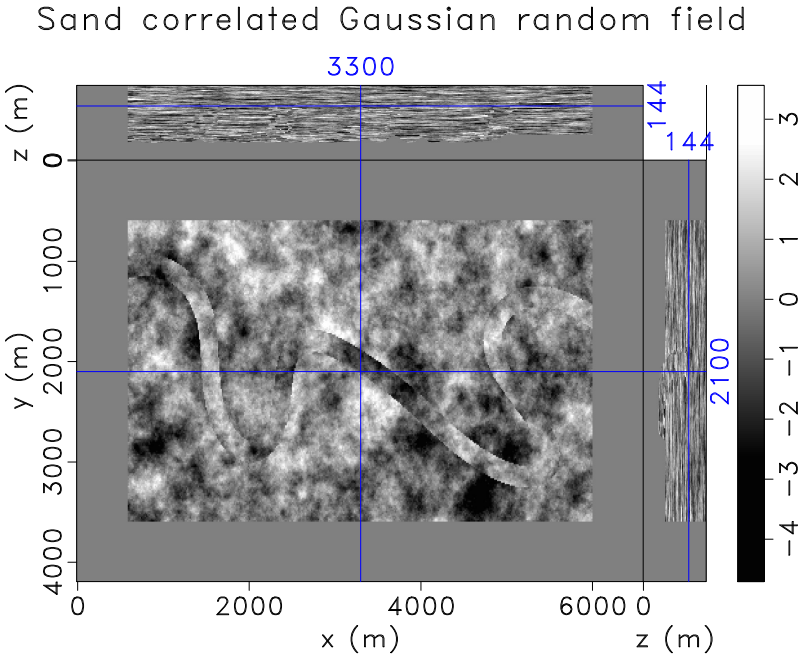


Figure 4.8: (a) Sand fraction. (b) Shale fraction. (c) Porosity. (d) Noisy porosity. ch04-training/./model res-sand-xypad,res-shale,res-phi,res-phi-noise

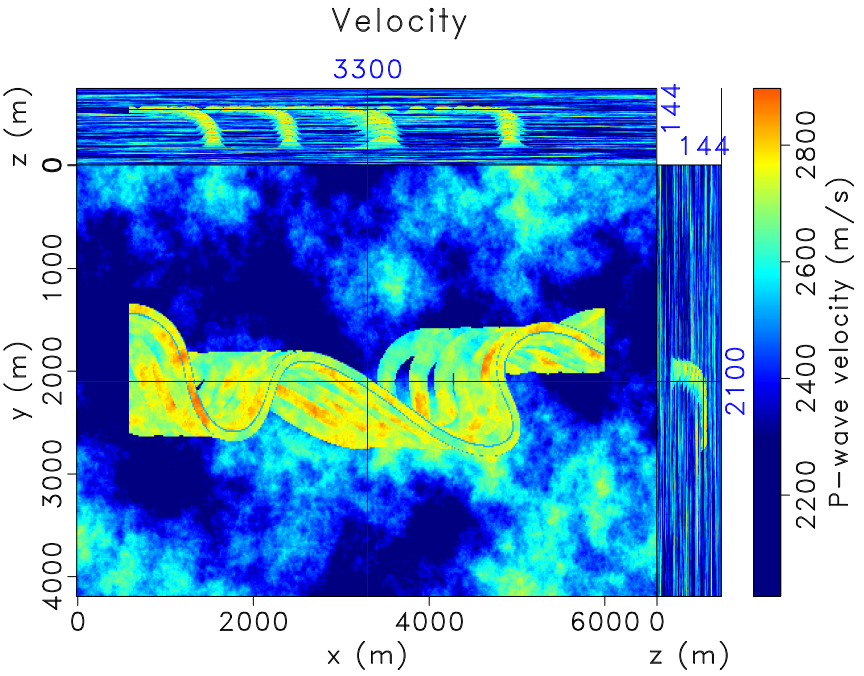


(a)

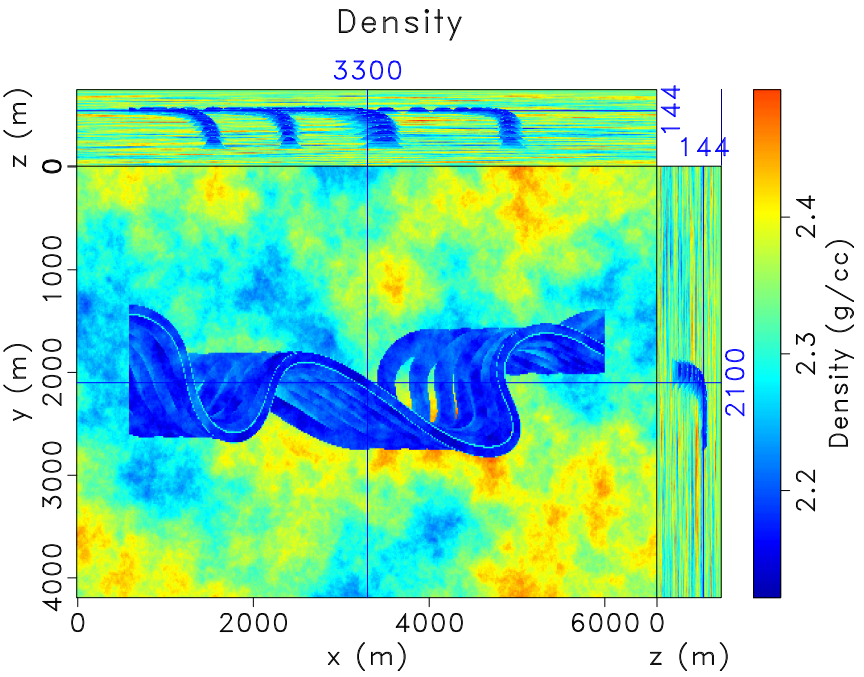


(b)

Figure 4.9: (a) Background noise realization. (b) Sand correlated Gaussian random field. ch04-training/./model res-bk-sim-01-real,sd-rfield

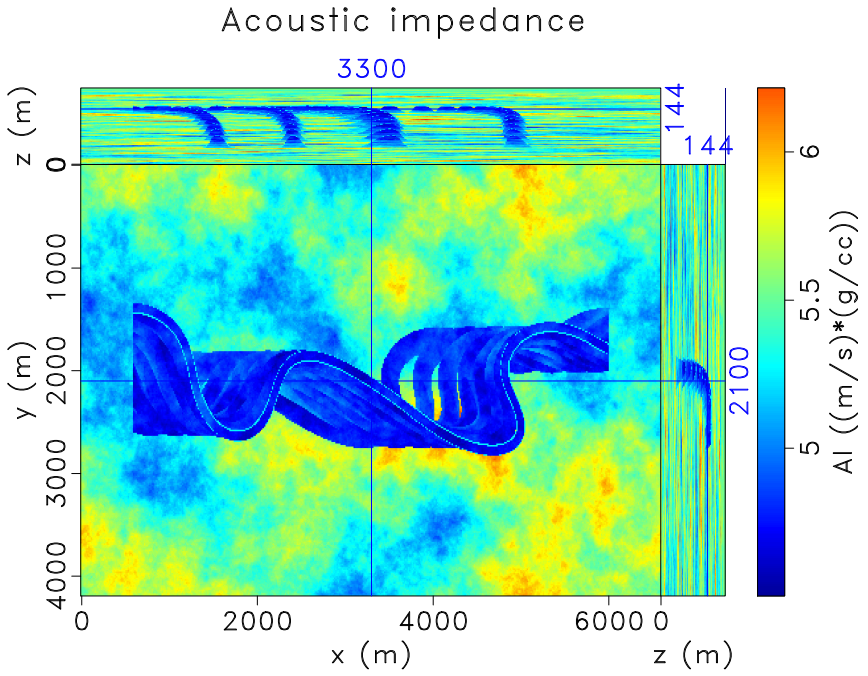


(a)

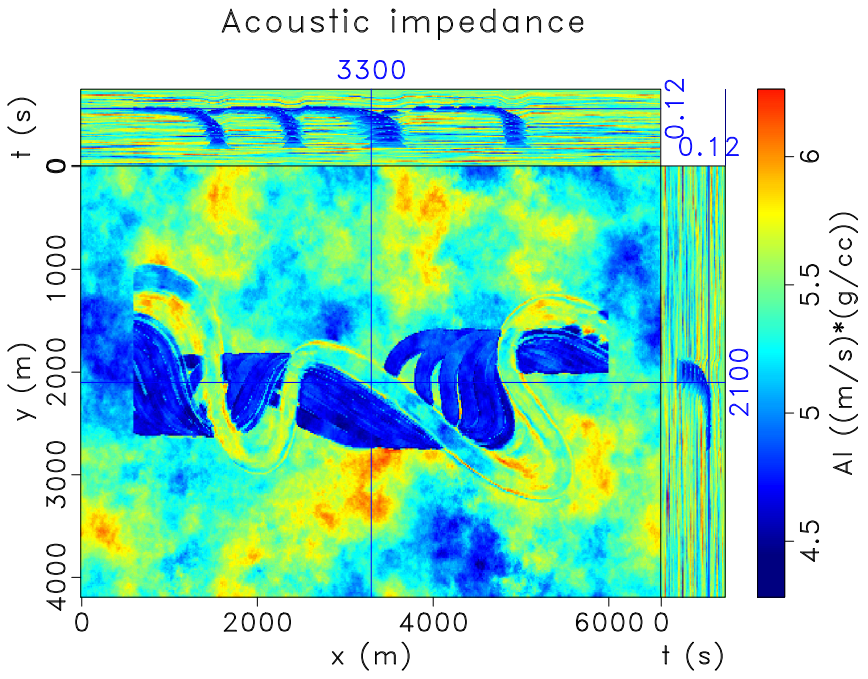


(b)

Figure 4.10: (a) P-wave velocity. (b) Density. ch04-training/./model vel,den



(a)



(b)

Figure 4.11: Acoustic impedance in (a) depth. (b) time. ch04-training/./model aim,ait

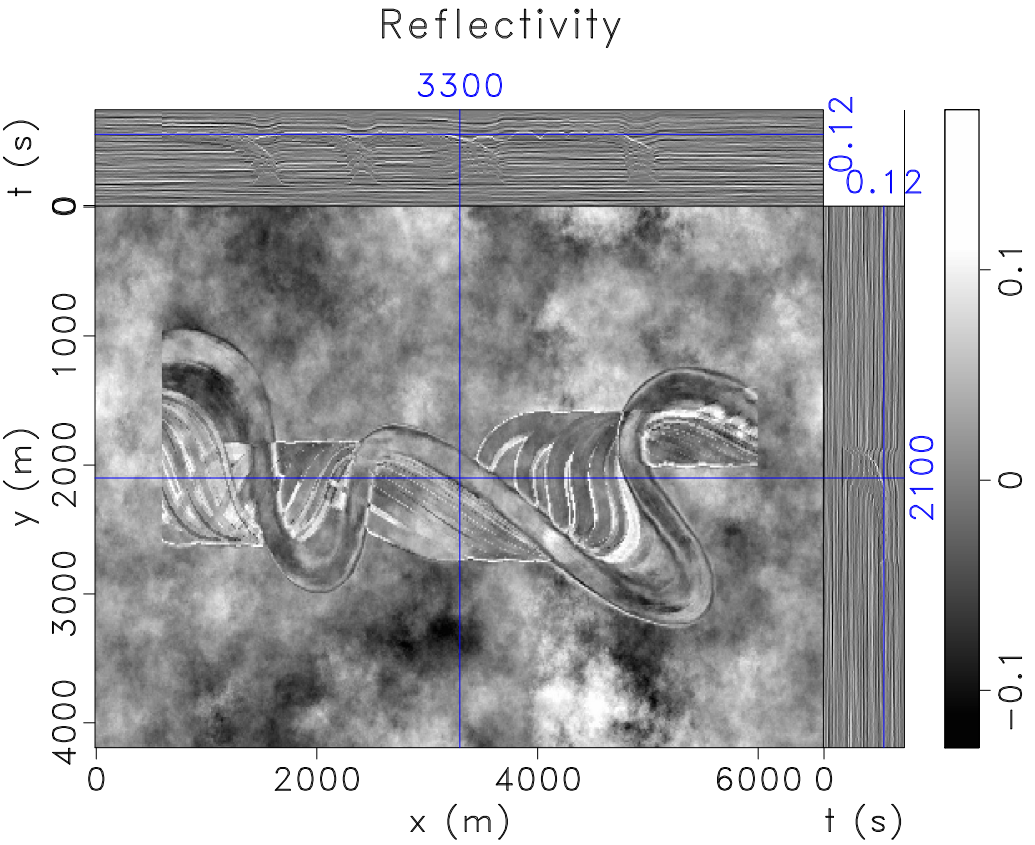


Figure 4.12: Reflectivity in time. ch04-training/./model ret

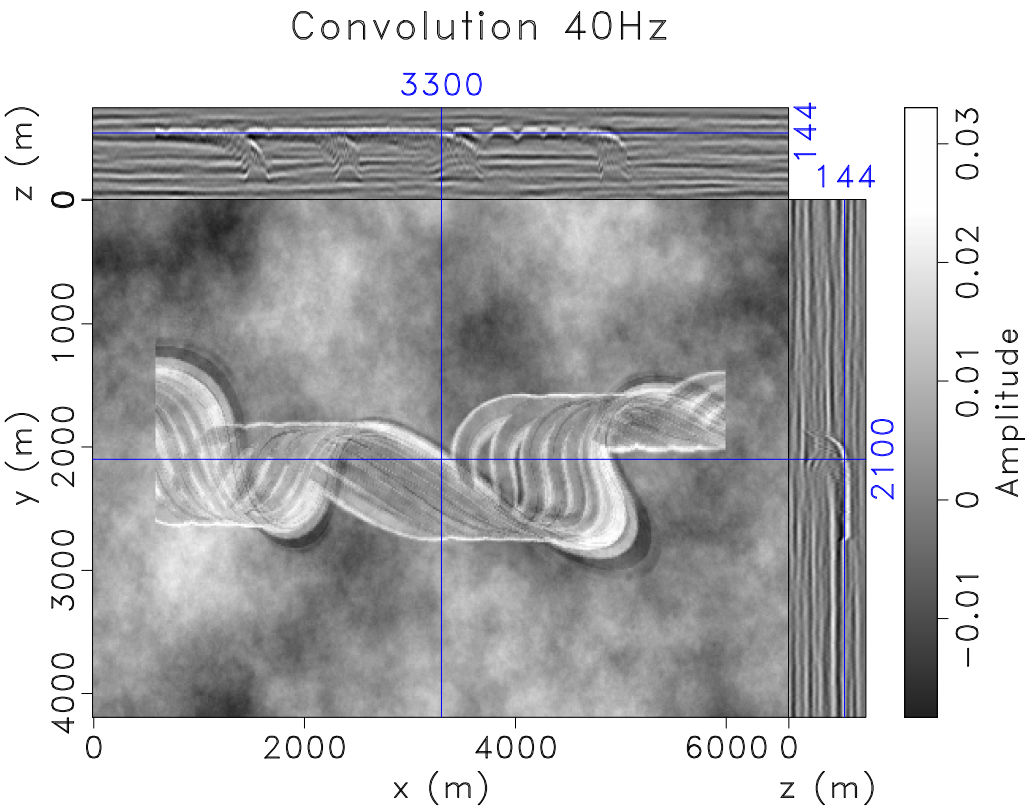


Figure 4.13: Convolutional depth model. ch04-training/./model mt3d-40

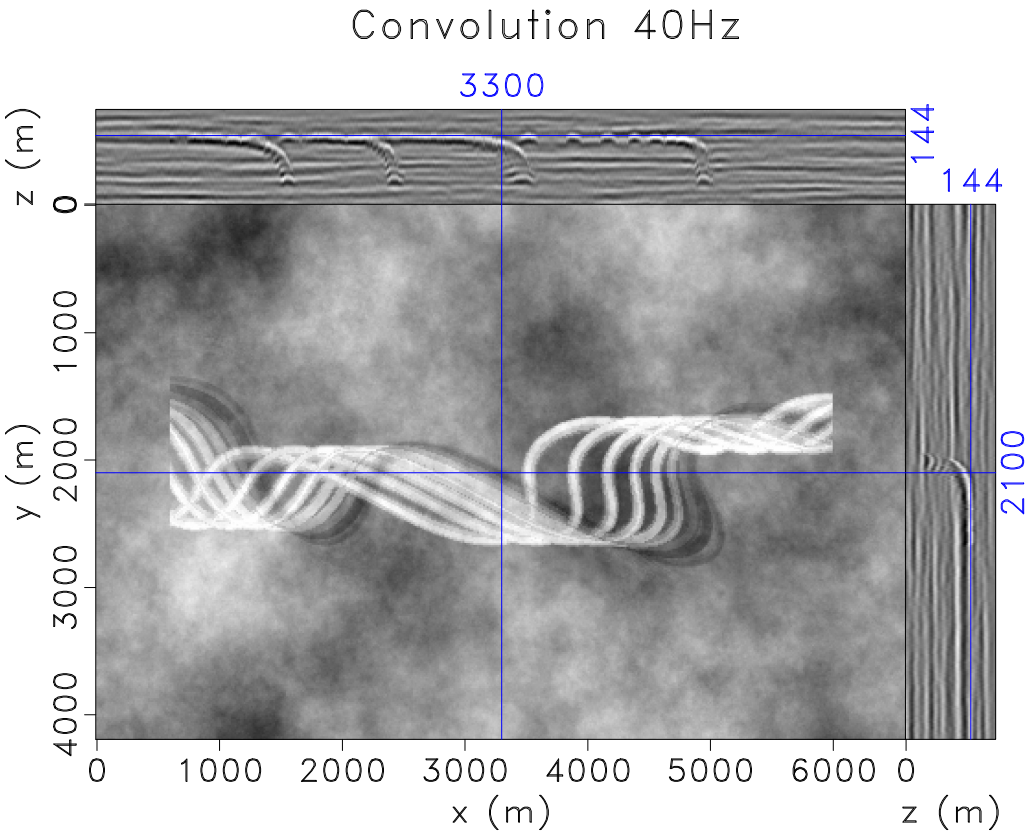
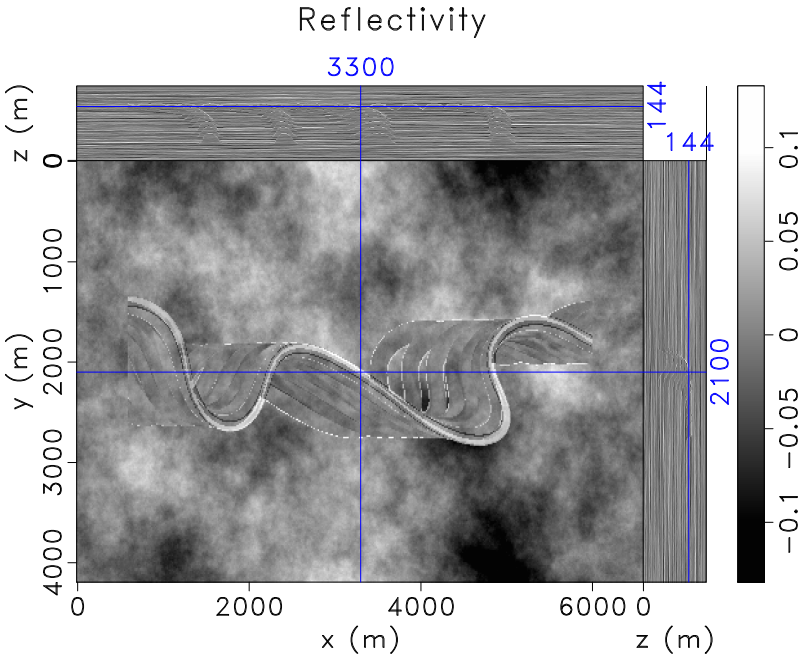
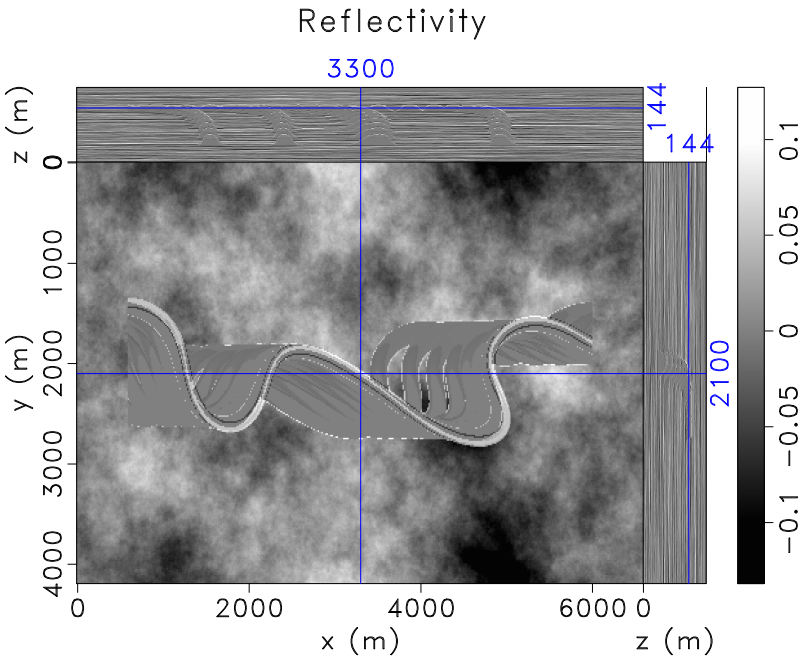


Figure 4.14: An example of synthetic training data with thin channels. ch04-training/./label mt3d-40-2

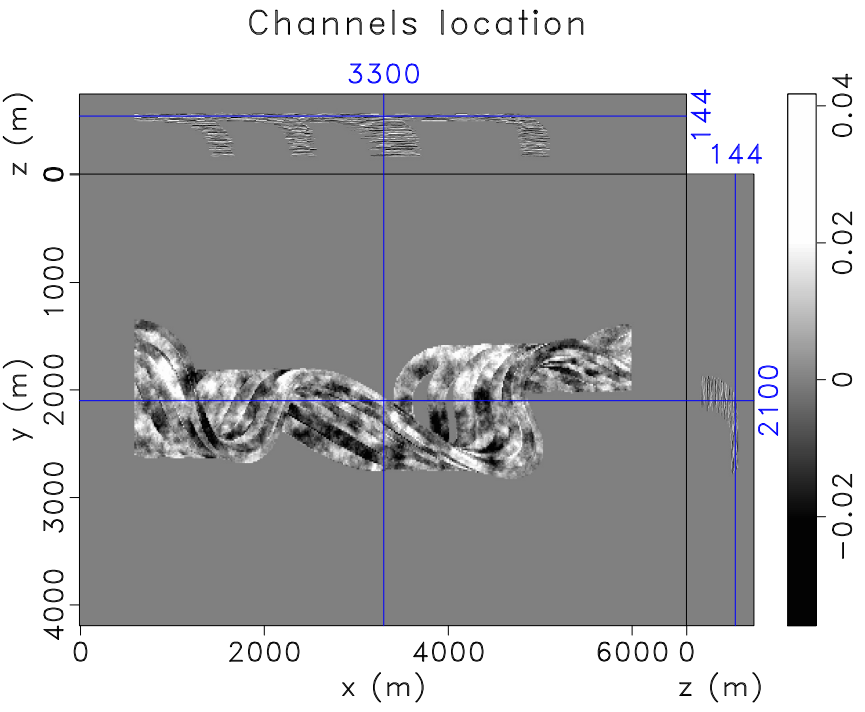


(a)

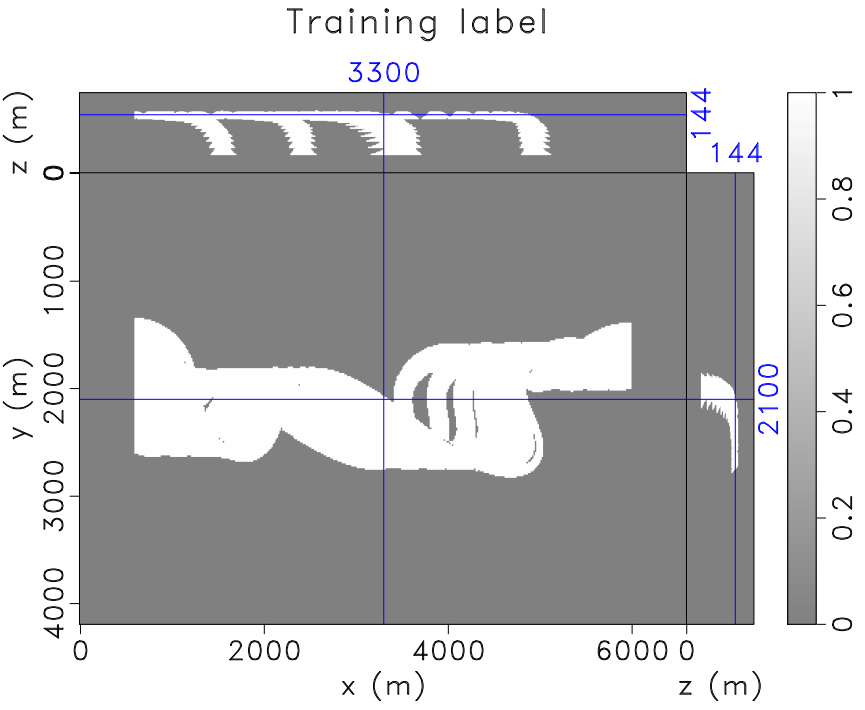


(b)

Figure 4.15: Reflectivity in depth (a) with noise. (b) after eliminating noise inside channels. ch04-training/./label ref,reflbl



(a)



(b)

Figure 4.16: (a) Channels location. (b) Training label. ch04-training/./label diff,label

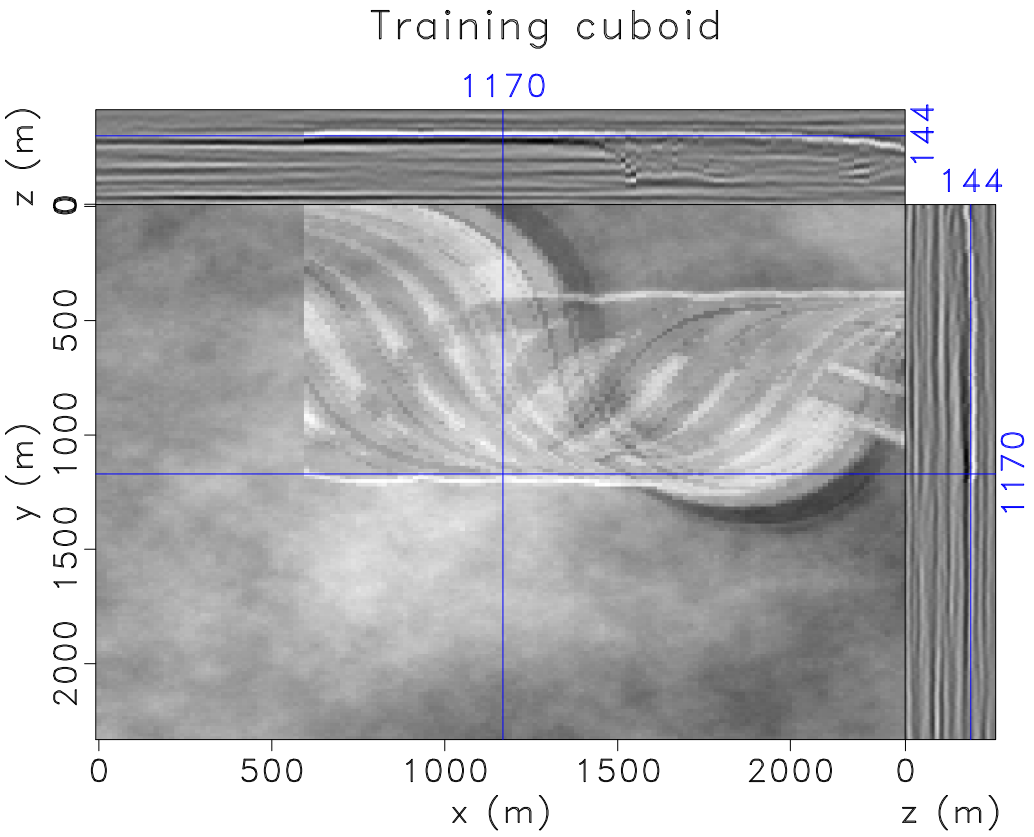


Figure 4.17: Training cuboid. ch04-training/./label prdw9

stages. My architecture for automatic channel detection has 16 filters with the size of $3 \times 3 \times 3$ in a convolutional layer. The weights W are initialized with a random orthognal matrix (Saxe et al., 2013). W is norm preserving $||W\,x||_2 = ||x||_2$, which helps to keep the norm of the input constant throughout the network at the start of training, so it prevents network from exploding and vanishing gradients problem. W has columns and rows which are all orthonormal to one another, which encourages the weights to learn different input features. The biases are initialized with 0. Each convolutional layer comes with a batch normalization layer to normalize the data and control overfitting (Ioffe and Szegedy, 2015). Non-linear activation function ReLU is inserted to learn non-linear relationships. Max-pooling layers with $2 \times 2 \times 2$ kernels are added in between each convolutional layer to reduce the spatial size of feature maps and control overfitting.

Each decoder layer upsamples the input feature maps and convolves the output with a trainable decoder filter to produce a dense map. Upsampling layers use transposed convolution algorithm with learnable $2 \times 2 \times 2$ filters (Figure 4.20). The coarse outputs are convolved with learnable $3 \times 3 \times 3$ filters to produce denser feature maps (Figure 4.21). The output from last decoder layer is fed into a $1 \times 1 \times 1$ convolutional layer to produce a feature map corresponding to two labels of channel or non-channel. The weight of this layer is initialized using MSRA initialization, which follows a zero-mean Gaussian distribution whose standard deviation is $\sqrt{\frac{2}{n_l}}$ where $n_l = k^2\,c$, k is the filter size, and c is the number of input feature maps (He et al., 2015). The last layer is softmax layer that produces the probabilities of each label for each pixel in the seismic volumes. To quantify the uncertainty of the model, I use a dropout layer between the last encoder layer and the first decoder layer, which removes 30% of the units. Adam optimizer (Kingma and Ba, 2014) with 0.1 as learning rate is used for

backpropagation. I take 30 samples at test time and calculate the variance of the distribution over the probabilities of channel to quantify the prediction uncertainty.

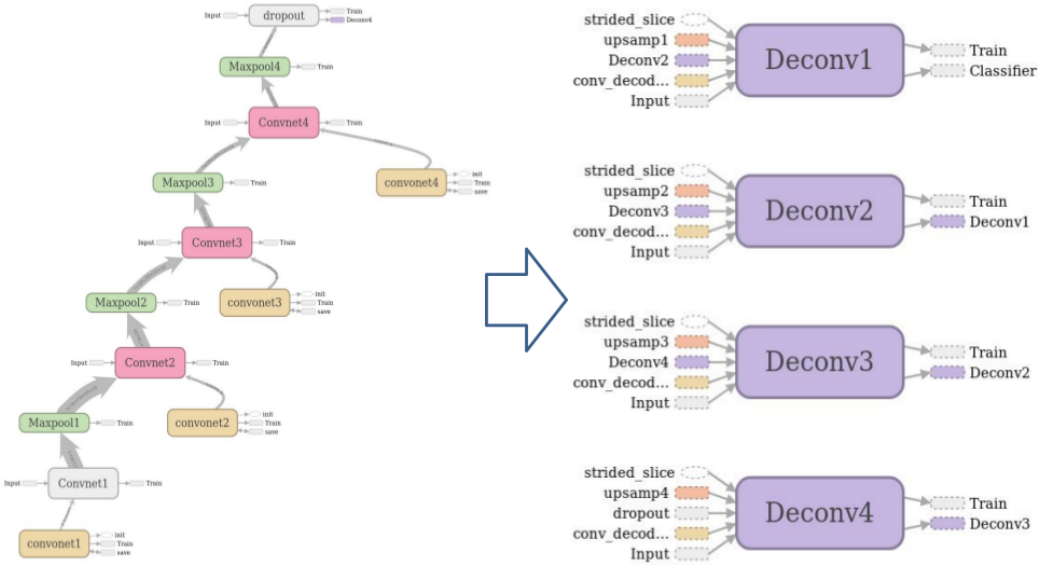


Figure 4.18: Encoder-decoder convolutional neural network architecture. ch04-training/./training architecture

TRAINING RESULTS

I trained the network on the synthetic data described above for 33 epochs in 4 hours using a Titan Xp GPU. The global accuracy defined as the percentage of pixels correctly classified in the image increases during training and reaches 99%. However, it is not a good metric to judge the network because in a seismic volume, there are few pixels with channel label compared to background label. Mean value of Intersection over Union (Mean IU) is the accuracy metric defined as

$$(\frac{1}{n_{cl}}) \sum_i \frac{n_{ii}}{\sum_j n_{ij} + \sum_j n_{ji} - n_{ii}}, \tag{4.10}$$

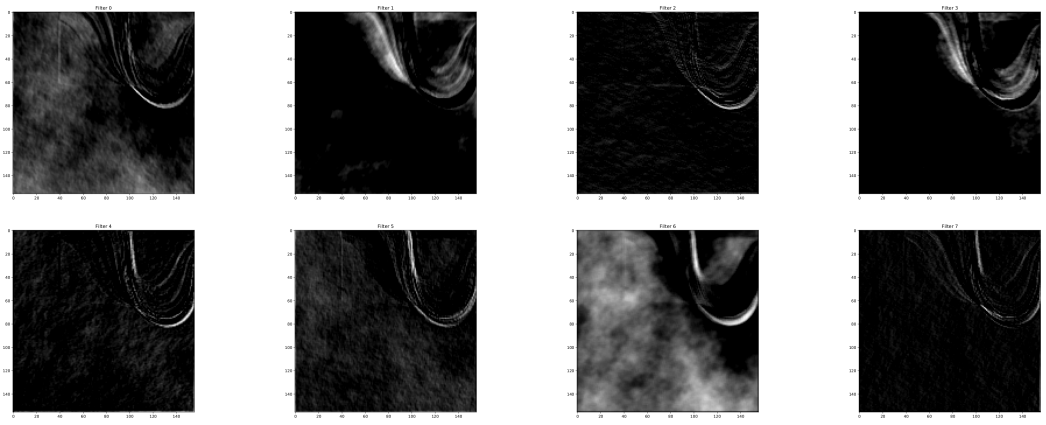


Figure 4.19: 8 example feature maps generated by a convolutional layer. ch04-training/./training conv1-1

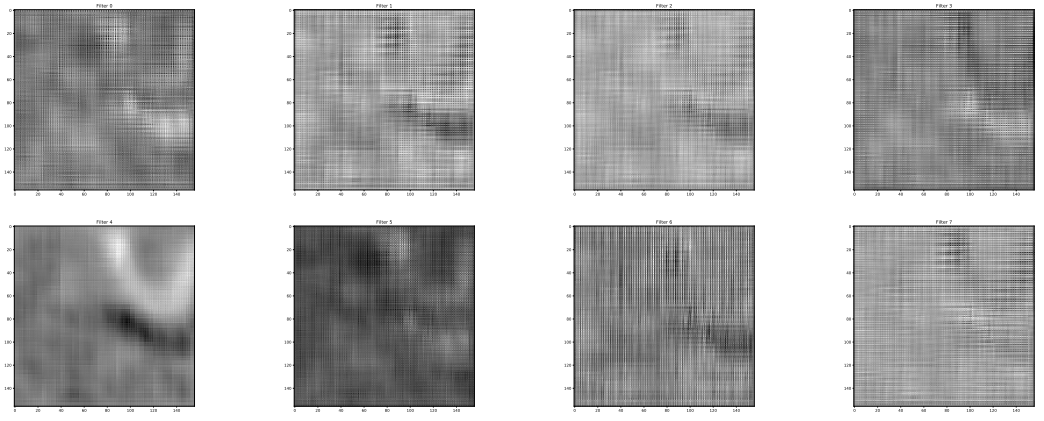


Figure 4.20: 8 example feature maps generated by transposed convolution upsample filters. ch04-training/./training upsamp1-1

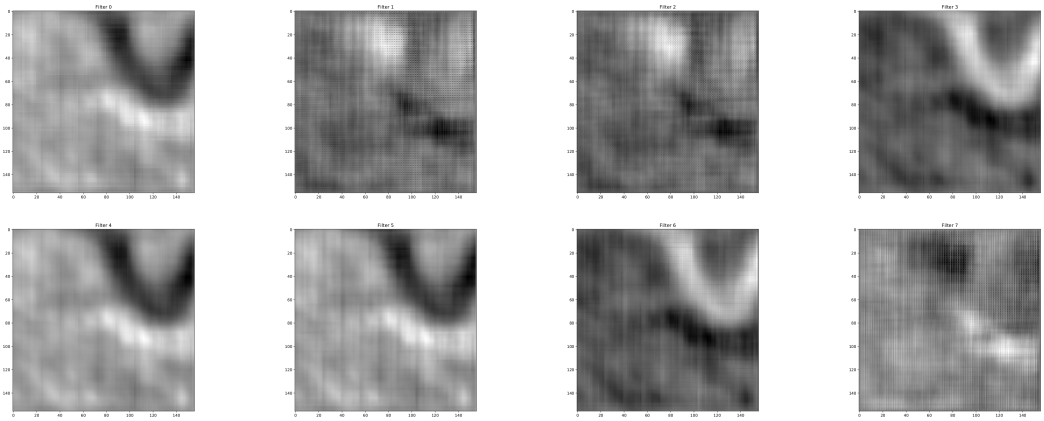


Figure 4.21: 8 example feature maps generated by denser upsample filters. ch04-training/./training convdecode1-1

where n_{cl} is number of classes, n_{ij} is the number of pixels of class i predicted to belong to class j (Long et al., 2014a). The cross-entropy cost decreases during training (Figure 4.22) and the mean IU is 93.5% after training. To evaluate the results, I extract a vertical slice with corresponding true label (Figure 4.23) and a horizontal slice with corresponding ground truth (Figure 4.24). The results show that channel bodies are picked clearly in the synthetic dataset (Figure 4.23(c)). The model uncertainty from using dropout layer can be used to understand how confidently we can trust the channel segmentation. At boundaries of channels, the prediction has high uncertainty (Figure 4.23(d)). It reflects the ambiguity of the network surrounding the definition of defining the transition between the channel and non-channel areas (Kendall et al., 2015). It is difficult to distinguish individual channels in the dataset (Figure 4.24(c)), so there is high uncertainty where there are multiple channels (Figure 4.24(d)). With this uncertainty volume, interpreters can repick the areas with high uncertainties to enhance the results from the neural network.

The method I propose for automatic channel detection in 3D seismic volumes

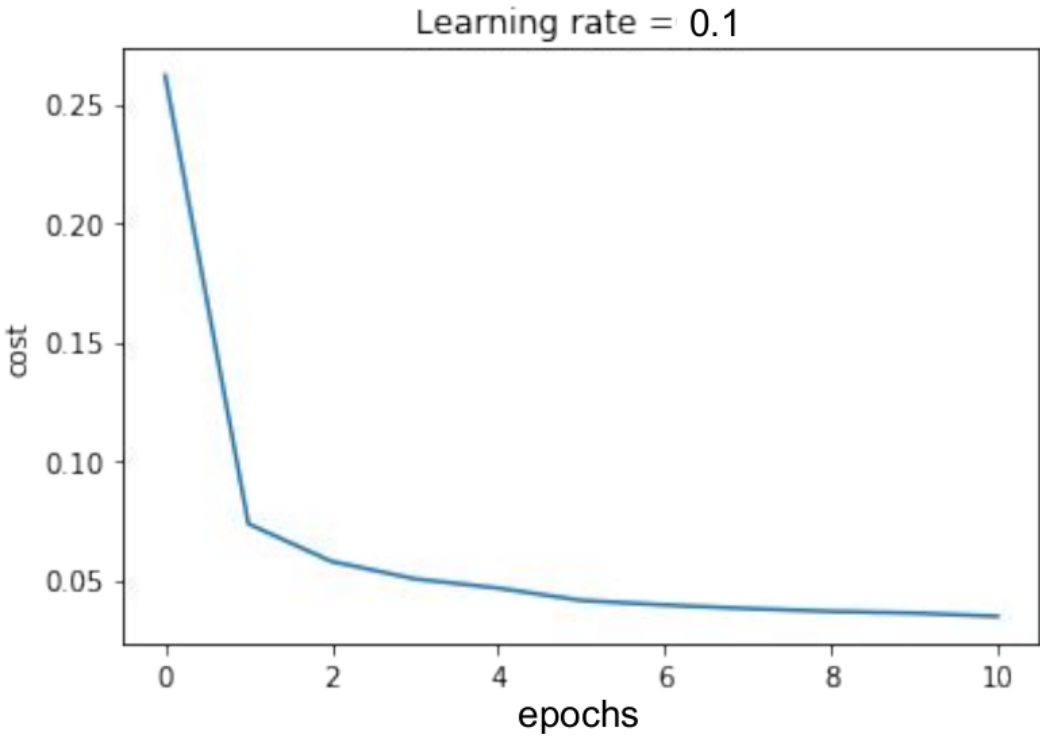
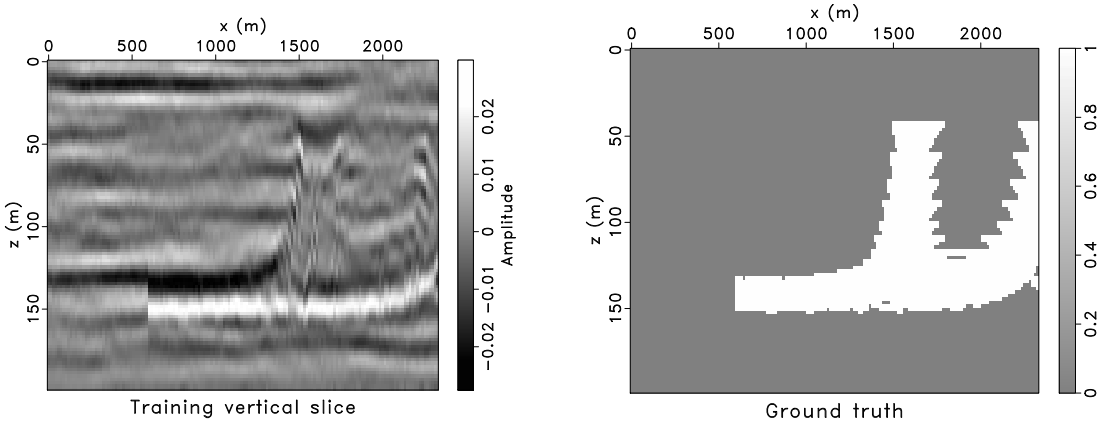


Figure 4.22: Training cost in 10 epochs. ch04-training/./training cost2



(a) (b)

(c) (d)

Figure 4.23: (a) Training vertial slice. (b) Ground truth of the training vertical slice. (c) Channel probability in the vertical slice. (d) Model uncertainty in the vertical slice. ch04-training/./training prdw9ver,prdw10ver,prdw7ver,vadw7ver

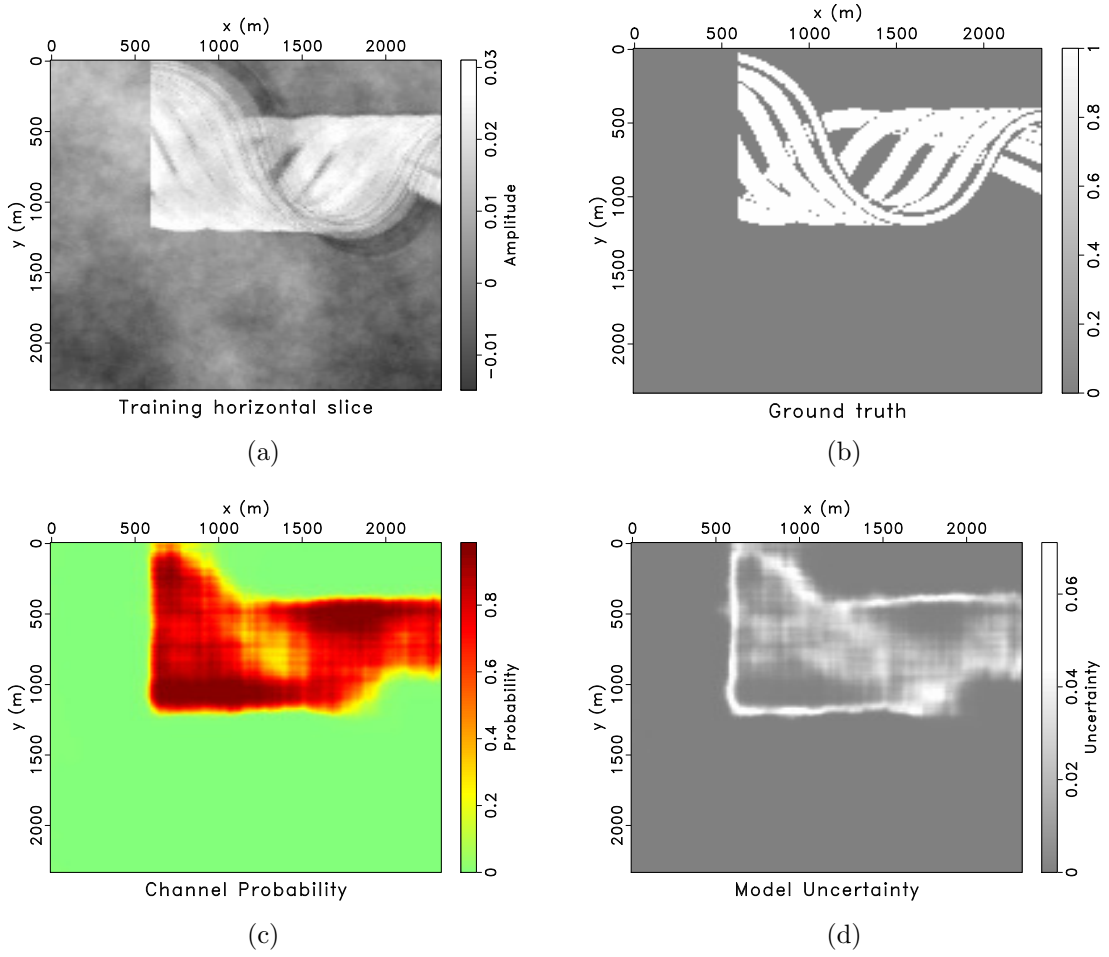


Figure 4.24: (a) Training horizontal slice. (b) Ground truth of the training horizontal slice. (c) Channel probability in the horizontal slice. (d) Model uncertainty in the horizontal slice. ch04-training/./training prdw9hor,prdw10hor,prdw7hor,vadw7hor

is based on an encoder-decoder convolutional neural network. I propose to train the network with a synthetic dataset. My proposed method generates two volumes of channel probabilities and model uncertainties at every pixel in the seismic volume. The synthetic dataset is created by geologists, geophysicists, and petroleum engineers. Using synthetic dataset helps the network get correct training label and is not biased to manual interpretations. However, synthetic dataset does not reflect all geologic conditions in real world like field dataset.

Chapter 5

Automatic channel detection using deep learning - Testing phase*

The trained model in Chapter 4 is applied directly to two 3D field datasets with different geologic conditions to test for the generalization. The first field dataset is a 3D marine seismic data in depth from Browse basin, offshore Australia (Figure 5.1). The seismic data hosts numerous stacked deep-water channel-levee complexes (Rosleff-Soerensen et al., 2012). The second field dataset is a 3D seismic data in time from Parihaka, New Zealand (Figure 5.2). The relative coarse-grained channel deposits are at the base of the incisional channel systems, which is different from the Australia dataset where the coarse-grained channel deposits are vertically stacked (Salazar et al., 2016).

Australia 3D marine seismic data

I apply the weights from training the synthetic data to a $312 \times 312 \times 100$ subvolume of a field dataset from offshore Australia. The dataset is a 3D marine seismic survey located in 2500 meters water depth with a sampling rate of 2 ms and a dominant frequency of 120 Hz. The dataset is in depth with sampling inter-

*Parts of this section are published in Pham et al. (2018, 2019). All co-authors contributed equally.

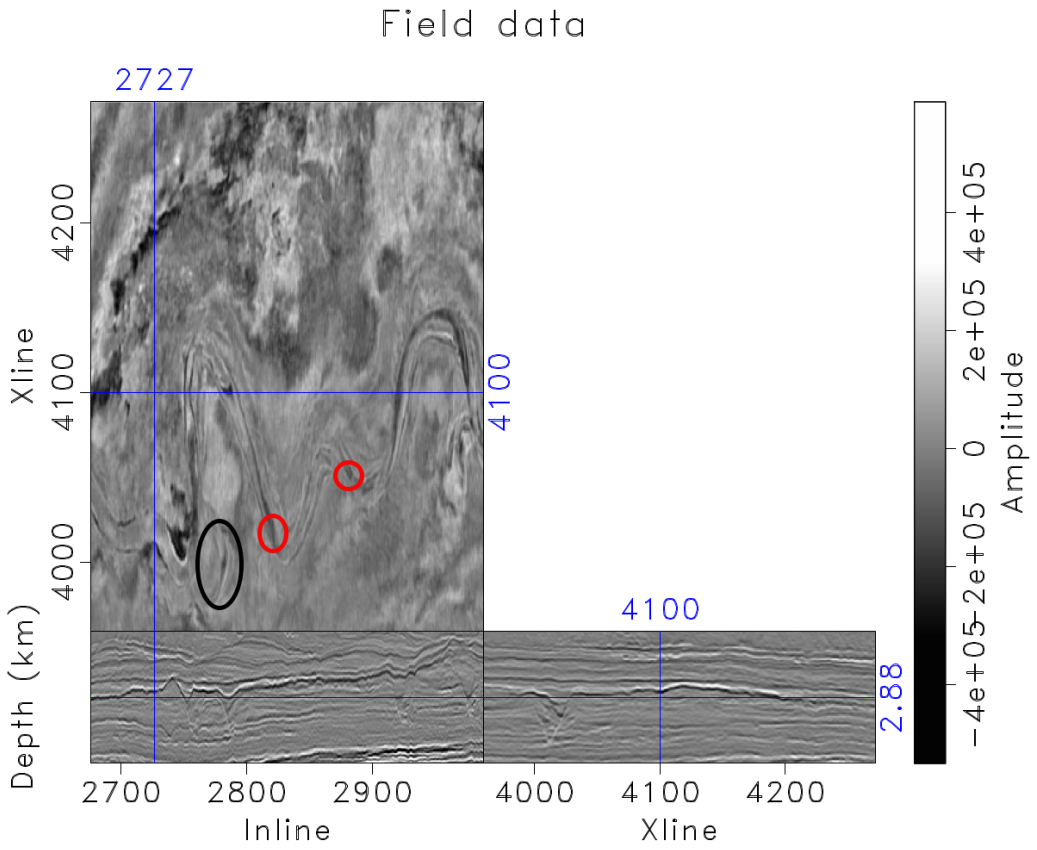


Figure 5.1: Australia field dataset (Red circles are thin channel areas. Black circles are multiple channels areas.). ch05-testing/../SEG imageausnew3

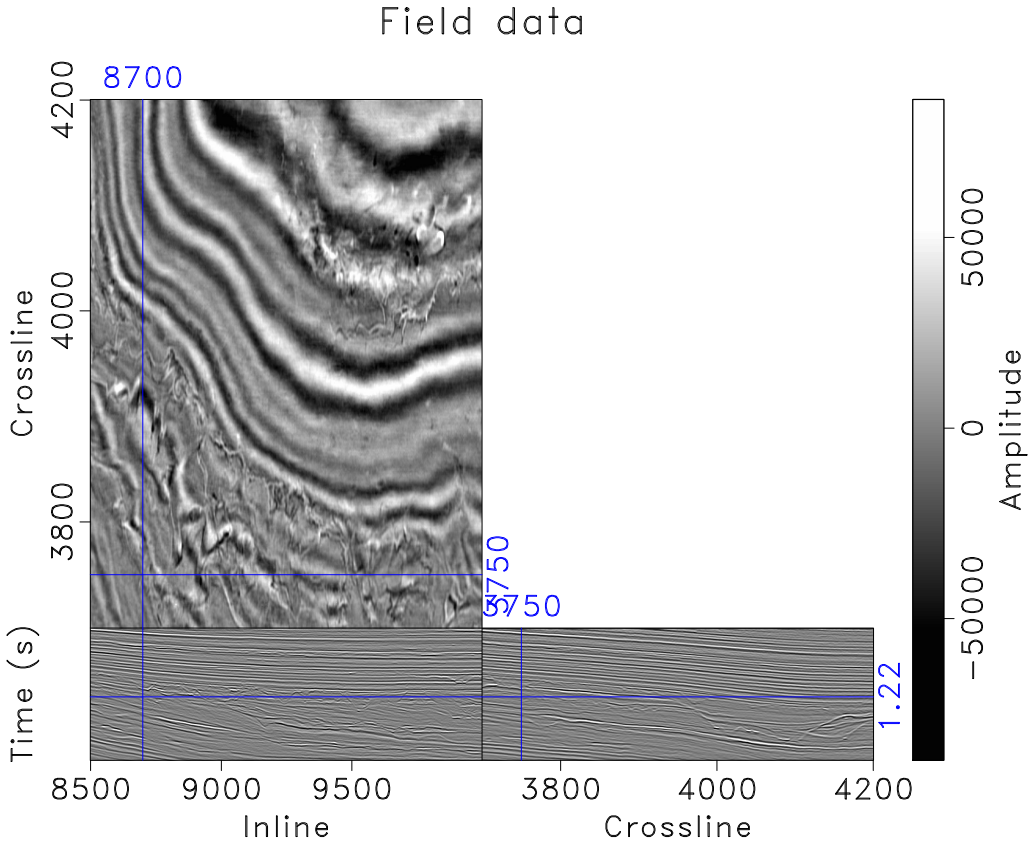
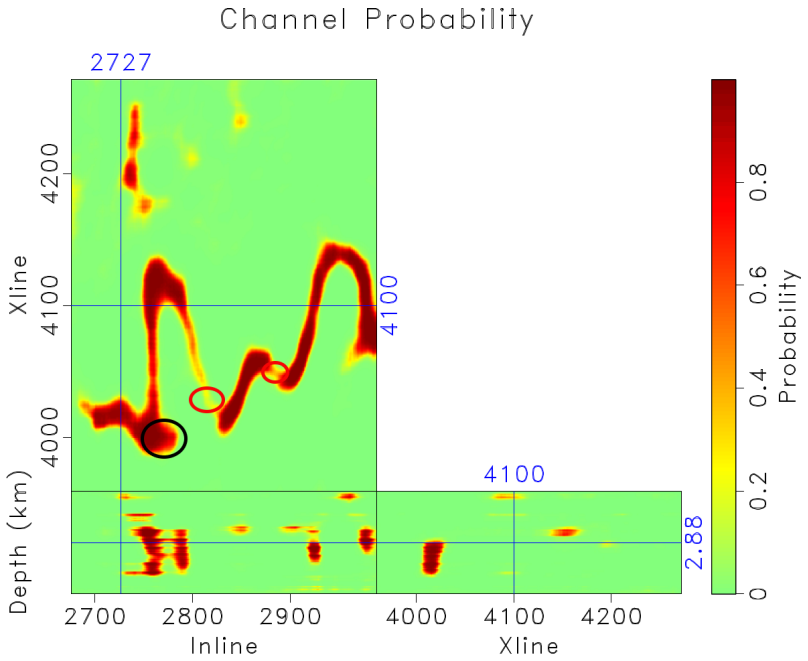


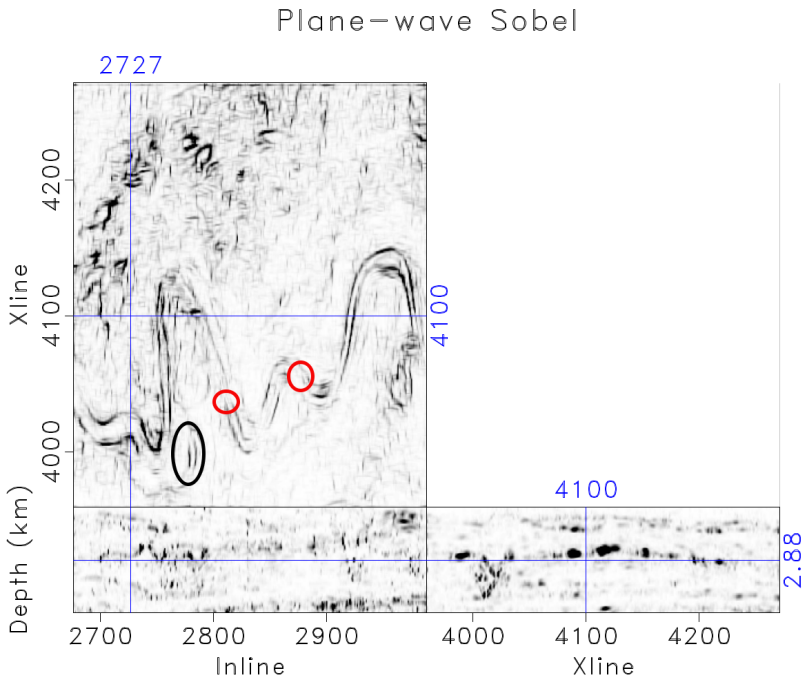
Figure 5.2: Parihaka field dataset. ch05-testing/../SEG imageParinew3

val of 2 meters. I divide the subvolume into 16 small overlapping volumes of size $156 \times 156 \times 100$ samples using nonstationary patching method (Claerbout, 2014) to eliminate the edge artifacts and test each volume independently. The testing output volumes are stitched together using the inverse of non-stationary patching method with weighted boundaries. Being only trained with synthetic dataset, the model can clearly pick the channel geobodies in the seismic volume (Figure 5.3(a)). Our result follows the channel edges enhanced by plane wave destruction Sobel filter (Phillips and Fomel, 2017) (Figure 5.3(b)), with the addition of model uncertainty. We analyze the prediction uncertainty by using the variance of 30 samples from the posterior distribution of channel probability (Figure 5.4).

When there are multiple channels in the dataset (black circle in Figure 5.1), the trained model cannot distinguish individual channels very well and the prediction uncertainty is high. The trained model can detect thin channels in the dataset with not too high probabilities, but the uncertainty map displays high values in these regions (red circles in Figure 5.1). Therefore, the prediction uncertainty has useful information for channels detection task and interpreters can use it to polish the detection result from neural network. The inference time for this dataset is only 3 minutes with a GPU, which is quicker than manual interpretation time. An horizon is manually picked following the channels in a cross-section of seismic data (Figure 5.5). The probability map (Figure 5.6) with organge color for high probabilities and blue color for low probabilities shows that the neural network results are following the manually picked horizon. However, there is a mispick in the black circle, but the uncertainty map (Figure 5.7) with the same color bar as probability map shows that there is high uncertainty in this region. Therefore, the network is not completely sure about its segmentation and the interpreters cannot trust the result in this area.



(a)



(b)

Figure 5.3: (a) Channel probability in the Australia field dataset. (b) Channel boundaries enhancement in the Australia dataset by PWD Sobel filter. (Red circles are thin channel areas. Black circles are multiple channels areas.). ch05-testing/../SEG ausnew3,aussobel3 79

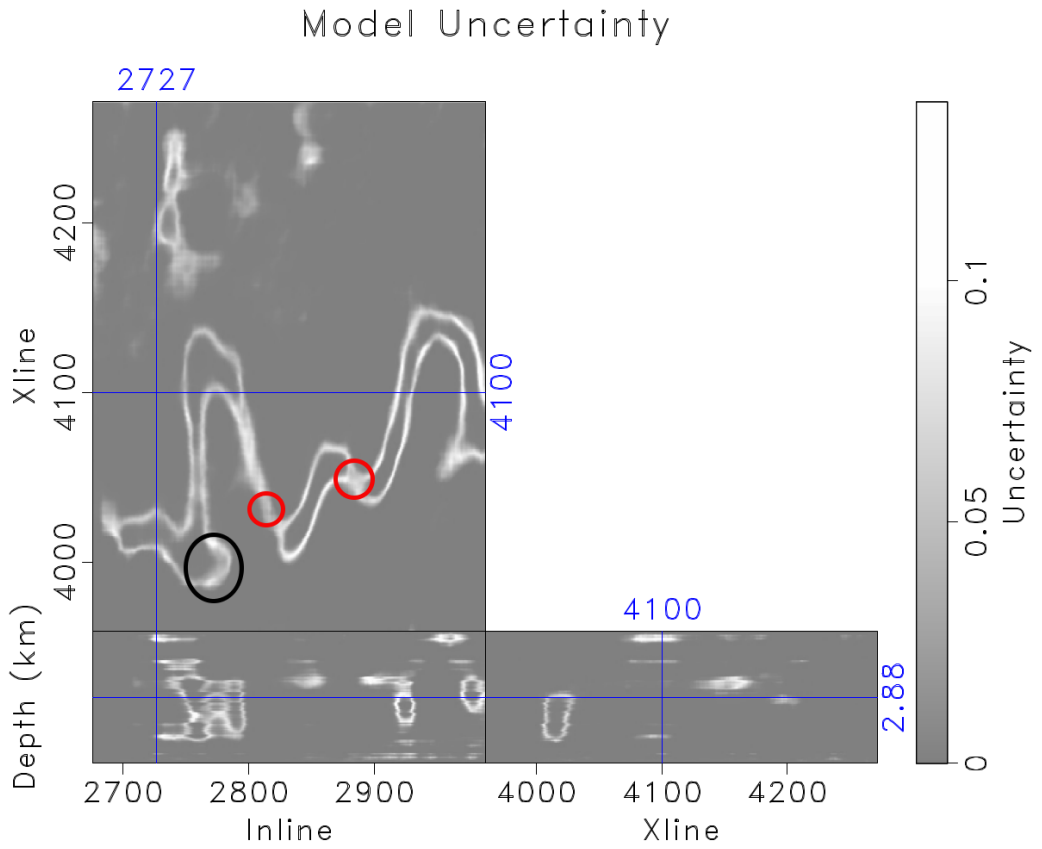


Figure 5.4: Model uncertainty in the Australia field dataset (Red circles are thin channel areas. Black circles are multiple channels areas.). ch05-testing/../SEG ausunnew3

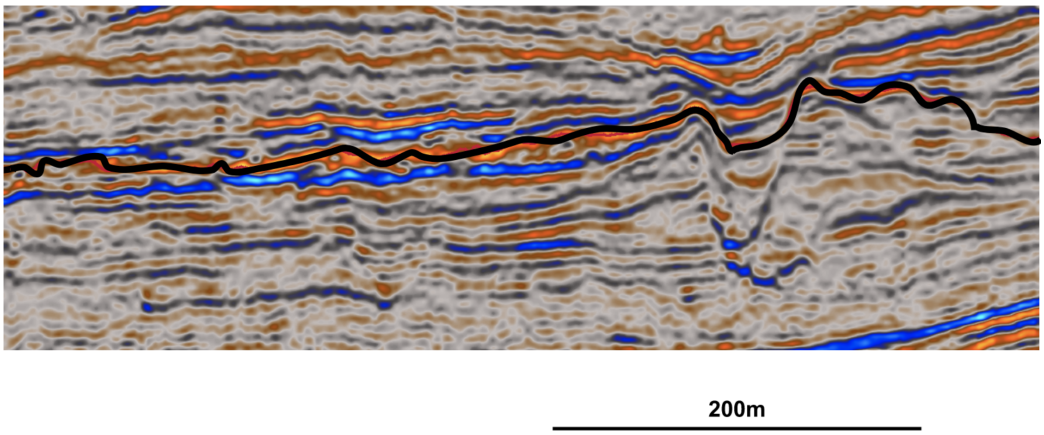


Figure 5.5: Manually picked horizon. ch05-testing/../SEG manpick

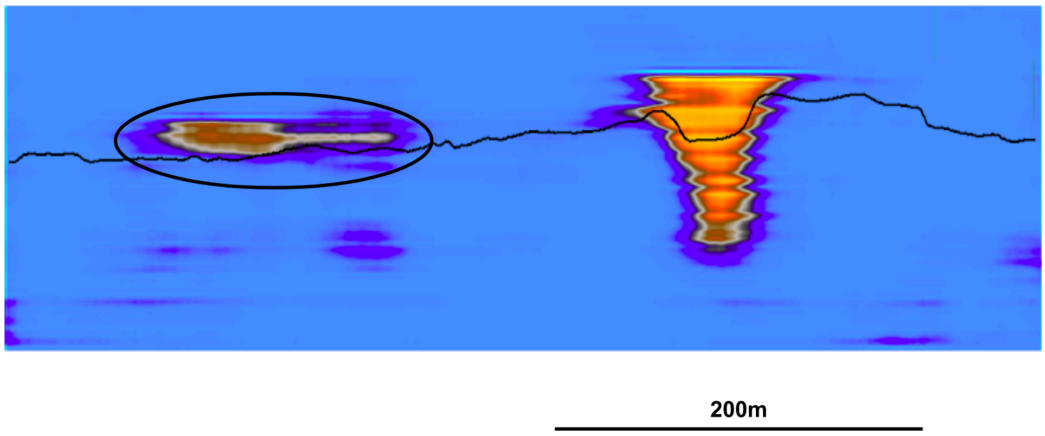


Figure 5.6: Channel probability from my method. ch05-testing/../SEG probman

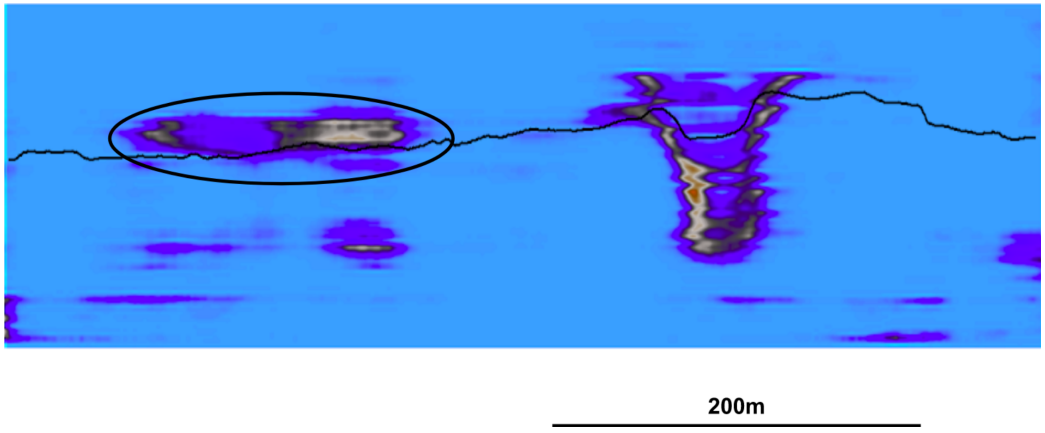


Figure 5.7: Model uncertainty from my method. ch05-testing/../SEG varman

Parihaka 3D seismic data

I apply the weights from training the synthetic data to a $501 \times 750 \times 251$ subvolume of the Parihaka dataset in New Zealand. The dataset is in time with a sampling rate of 4 ms. We also use nonstationary patching method (Claerbout, 2014) to divide the subvolume into small overlapping volumes of size $156 \times 156 \times 100$ samples in order to eliminate edge artifacts. Being only trained with synthetic dataset, the model can clearly pick the channel geobodies in the seismic volume (Figure 5.8). The model uncertainty is calculated by using the variance of 30 samples from the posterior distribution of channel probability (Figure 5.9). Parihaka dataset is different from our synthetic training dataset so applying our trained model is hard to produce a clean probability volume.

Comparing with the results from seismic attributes discussed in Chapter 3, the results are comparable with the addition of an uncertainty volume. My proposed method does not require any pre-calculated seismic attributes for channel detection. It picks and isolates channel areas in seismic volumes with assigned probabilities. The produced probability volume is cleaner than the detection results from seismic attributes. Calculating seismic attributes, for example, the eigen-decomposition-based coherence takes three days. The inference time with 30 sample Monte-Carlo simulation for this dataset is only 33 minutes with a GPU. Moreover, my proposed method also produces a way to quantify the model uncertainty. With this uncertainty volume, interpreters can enhance the results from neural network where the uncertainties are high. The uncertainty volume can be combined with the probability volume to calculate different realizations of reservoir volume. Therefore, it can help evaluate uncertainty of decision-making based on interpretations.

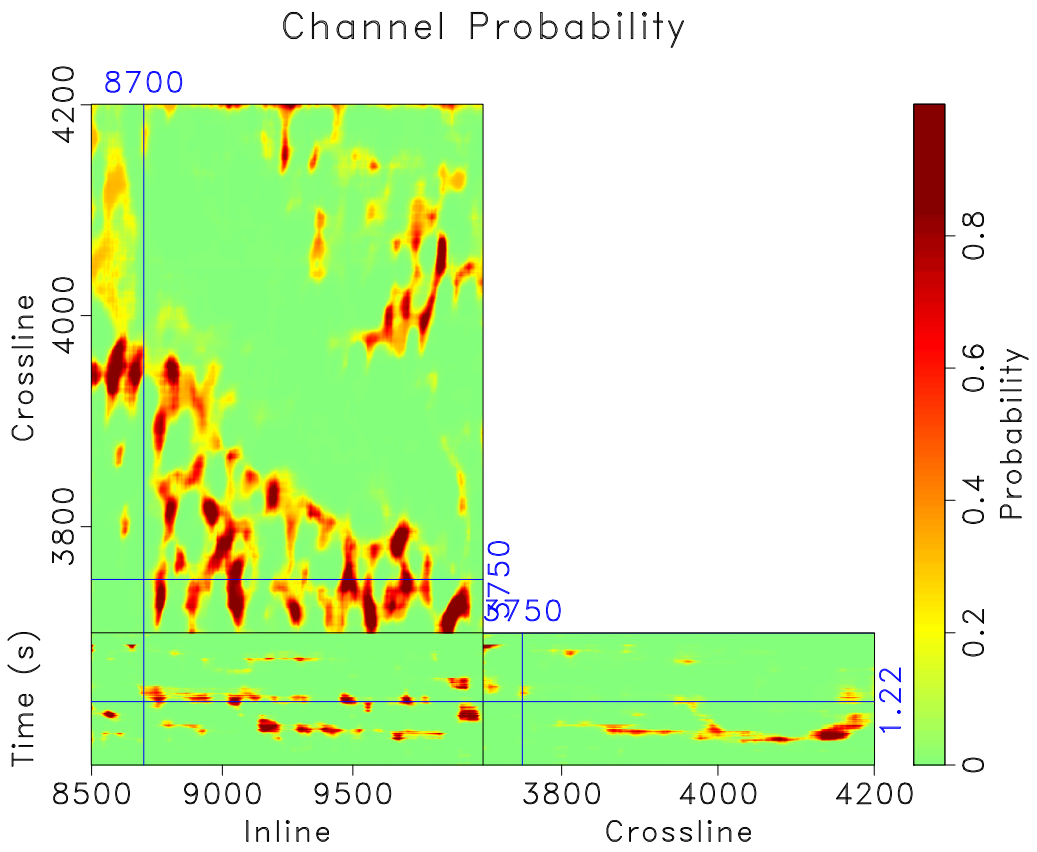


Figure 5.8: Channel probability in the Parihaka field dataset. ch05-testing/../SEG Parinew3

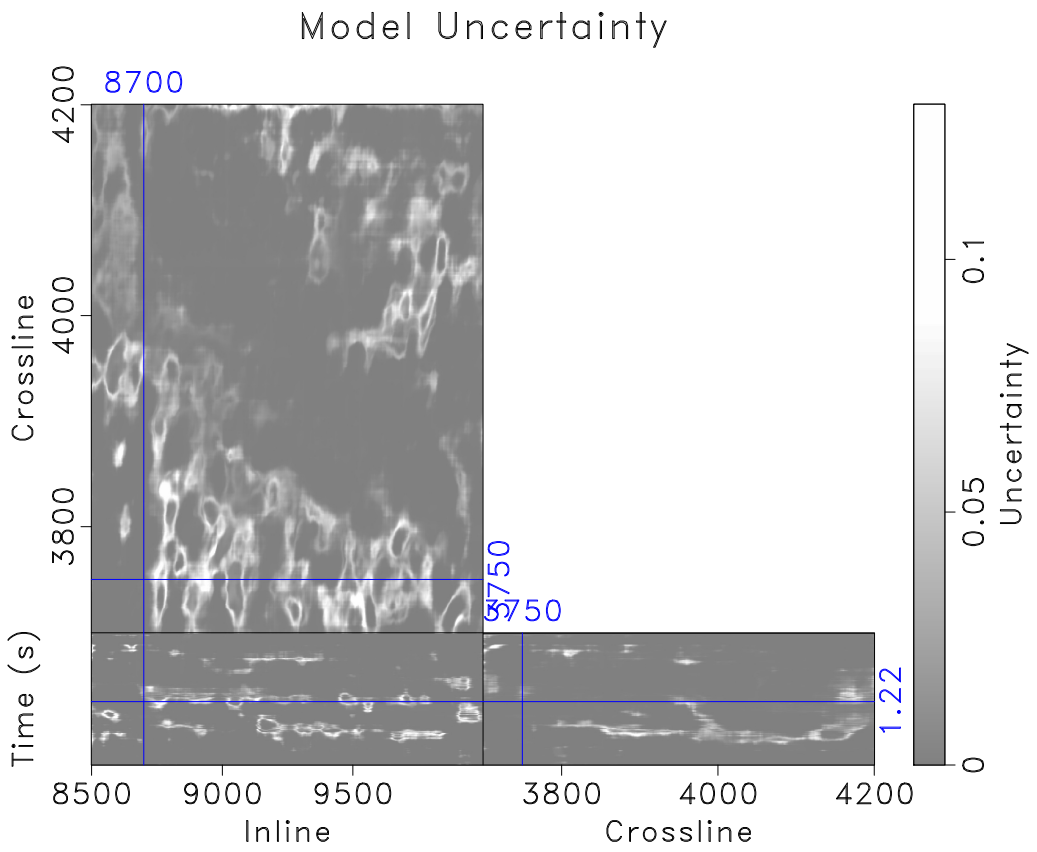


Figure 5.9: Model uncertainty in the Parihaka field dataset. ch05-testing/../SEG Pariunnew3

The neural network is only trained with the synthetic depth volumes, but it can successfully detect the channel geobodies in both time and depth 3D seismic volumes with different geologic conditions. The model uncertainty is helpful for interpreters in judging the results produced by neural network.

Chapter 6

Conclusions

Channels are important geologic features for hydrocarbon exploration. Detection of 3D channel geobodies in seismic volumes are time-consuming and subjective. Machine learning and deep learning are developing rapidly because of abundant available data and more powerful computational resources. With various types of data in geophysics, machine learning and deep learning are promising in exploration geophysics, especially in interpretations. I propose a workflow for automatic channel detection in 3D seismic volumes based on an encoder-decoder convolutional neural network. I propose to train the network with a synthetic dataset and apply the trained model directly on the field datasets. I test the proposed method by using a synthetic dataset obtained from the knowledge of geologists, geophysicists, petroleum engineers for training and two 3D seismic volumes in different domain, with different geologic conditions for testing. The method successfully detects the location of channels in the 3D seismic volumes, which is showed in the probability volume of channel at every pixel in the seismic volumes. Moreover, my method can also produce the model uncertainty volume by using dropout layer at test time.

My method does not need any pre-computed seismic attributes like conventional methods and still produces comparable results. It actually pick the 3D channel geobodies, which cannot be done easily by conventional methods. Moreover, the

proposed method only requires training with synthetic dataset and is fast in inference time. The probability volume does not detect the thin channels and multiple overlapped channels quite well. However, using the uncertainty volume produced by my proposed method, interpreters can enhance the detection results from neural network. It can also be combined with probability volume to get different realizations of reservoir volume.

Future work

Results of my proposed method for automatic channel detection in 3D seismic volumes can be improved by getting more diverse training dataset. The field datasets with manual intrepretations might be helpful in accounting for real world situations. Future research can incorporate segmentation with object detection (He et al., 2017) to clearly picked individual channels in the datasets. My proposed method is not limited only for channel features, it can also be used to detect faults or salt bodies in seismic volumes. It might be interesting to create a multi-class training dataset, where there is a combination of different geologic features.

Bibliography

Allen, J. G., 1977, Short term spectral analysis, synthetic and modification by discrete Fourier transform: IEEE Transactions on Signal Processing, 135–238.

Aqrawi, A. A., 2014, Adaptive Sobel based edge detection for enhaced fault segmentation: International Petroleum Technology Conference, 1424–1428.

Aqrawi, A. A., and T. H. Boe, 2011, Improved fault segmentation using a dip guided and modified 3D Sobel filter: 86th Annual International Meeting, SEG Expanded Abstracts, Society of Exploration Geophysicists, 999–1003.

Aqrawi, A. A., T. H. Boe, S. Barros, et al., 2011, Detecting salt domes using a dip guided 3D Sobel seismic attribute: 80th Annual International Meeting, SEG Expanded Abstracts, Society of Exploration Geophysicists, 1014–1018.

Badrinarayanan, V., A. Kendall, and R. Cipolla, 2015, Segnet: A deep convolutional encoder-decoder architecture for image segmentation: CoRR, **abs/1511.00561**.

Bahorich, M., and S. Farmer, 1995, 3-d seismic discontinuity for faults and stratigraphic features: The coherence cube: The Leading Edge, **14**, 1053–1058.

Boser, B. E., I. M. Guyon, and V. N. Vapnik, 1992, A training algorithm for optimal margin classifiers: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM, 144–152.

Calderón-Macías, C., M. K. Sen, and P. L. Stoffa, 1997, Hopfield neural networks, and mean field annealing for seismic deconvolution and multiple attenuation: GEOPHYSICS, **62**, 992–1002.

Chen, Z., S. Fomel, and W. Lu, 2013, Accelerated plane-wave destruction: Geophysics, **78**, V1–V9.

Chopra, S., 2002, Coherence cube and beyond: First Break, **20**.

Chopra, S., R. Kumar, and K. J. Marfurt, 2014, Seismic discontinuity attributes and Sobel filtering: 84th Annual International Meeting, SEG Expanded Abstracts, Society of Exploration Geophysicists, 1624–1628.

Claerbout, J., 2014, Geophysical image estimation by example.

Di, H., M. Shafiq, and G. AlRegib, 2018, Multi-attribute k-means clustering for salt-boundary delineation from three-dimensional seismic data: Geophysical Journal International, **215**, 1999–2007.

Dimitrakopoulos, R., and X. Luo, 1994, *in* Spatiotemporal Modelling: Covariances and Ordinary Kriging Systems: 88–93.

Dumoulin, V., and F. Visin, 2016, A guide to convolution arithmetic for deep learning: CoRR, **abs/1603.07285**.

Fomel, S., 2002, Applications of plane-wave destruction filters: Geophysics, **67**, 1946–1960.

———, 2007, Shaping regularization in geophysical-estimation problems: Geophysics, **72**, R29–R36.

———, 2009, Adaptive multiple subtraction using regularized nonstationary regression: GEOPHYSICS, **74**, V25–V33.

———, 2013, Seismic data decomposition into spectral components using regularized nonstationary autoregression: GEOPHYSICS, **78**, O69–O76.

Fomel, S., E. Landa, and M. T. Taner, 2007, Poststack velocity analysis by separation and imaging of seismic diffractions: GEOPHYSICS, **72**, U89–U94.

Fukushima, K., 1980, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position: Biological Cybernetics, **36**, 193–202.

Gal, Y., and Z. Ghahramani, 2015a, Bayesian convolutional neural networks with bernoulli approximate variational inference: CoRR, **abs/1506.02158**.

———, 2015b, Dropout as a bayesian approximation: Representing model uncertainty in deep learning: arXiv preprint arXiv:1506.02142.

Gersztenkorn, A., and K. J. Marfurt, 1999, Eigenstructure-based coherence computations as an aid to 3-d structural and stratigraphic mapping: GEOPHYSICS, **64**, 1468–1479.

Ghahramani, Z., 2015, Probabilistic machine learning and artificial intelligence: Nature, **521**, 452–459.

Guitton, A., H. Wang, and W. Trainor-Guitton, 2017, *in* Statistical imaging of faults in 3D seismic volumes using a machine learning approach: 2045–2049.

Hart, B. S., 2008, Channel detection in 3-d seismic data using sweetness: AAPG Bulletin, **92**, 733.

He, K., G. Gkioxari, P. Dollár, and R. B. Girshick, 2017, Mask R-CNN: CoRR, **abs/1703.06870**.

He, K., X. Zhang, S. Ren, and J. Sun, 2015, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification: CoRR, **abs/1502.01852**.

Hochreiter, S., and J. Schmidhuber, 1995, Long short-term memory.

———, 1997, Long short-term memory: Neural Comput., **9**, 1735–1780.

Hopfield, J. J., and D. W. Tank, 1985, "neural" computation of decisions in optimization problems: Biological Cybernetics, **52**, 141–152.

Ioffe, S., and C. Szegedy, 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift: CoRR, **abs/1502.03167**.

Janson, X., and S. Fomel, 2011, 3-d forward seismic model of an outcrop-based geocellular model.

Kendall, A., V. Badrinarayanan, and R. Cipolla, 2015, Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding: CoRR, **abs/1511.02680**.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: CoRR, **abs/1412.6980**.

Kington, J., 2015, Semblance, coherence, and other discontinuity attributes: The Leading Edge, **34**, 1510–1512.

Koza, J. R., F. H. Bennett, D. Andre, and M. A. Keane, 1996, *in* Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming: Springer Netherlands, 151–170.

Langley, P., 2011, The changing science of machine learning: Mach. Learn., **82**, 275–279.

LeCun, Y., Y. Bengio, and G. E. Hinton, 2015, Deep learning: Nature, **521**, 436–444.

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, 1989, Backpropagation applied to handwritten zip code recognition: Neural Comput., **1**, 541–551.

Lin, T., M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, 2014, Microsoft COCO: common objects in context: CoRR, **abs/1405.0312**.

Liu, G., S. Fomel, and X. Chen, 2011, Time-frequency analysis of seismic data using local attributes: GEOPHYSICS, **76**, P23–P34.

Liu, J., and K. J. Marfurt, 2005, *in* Matching pursuit decomposition using Morlet wavelets: 786–789.

———, 2007, Instantaneous spectral attributes to detect channels: GEOPHYSICS, **72**, P23–P31.

Liu, Y., and S. Fomel, 2013, Seismic data analysis using local time-frequency decomposition: Geophysical Prospecting, **61**, 516–525.

Long, J., E. Shelhamer, and T. Darrell, 2014a, Fully convolutional networks for semantic segmentation: CoRR, **abs/1411.4038**.

Long, J., N. Zhang, and T. Darrell, 2014b, Do convnets learn correspondence?: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1, MIT Press, 1601–1609.

Luo, Y., W. G. Higgs, and W. S. Kowalik, 1996, Edge detection and stratigraphic analysis using 3D seismic data: 76th Annual International Meeting, SEG Expanded Abstracts, Society of Exploration Geophysicists, 324–327.

Marfurt, K. J., R. L. Kirlin, S. L. Farmer, and M. S. Bahorich, 1998, 3-d seismic attributes using a semblance-based coherency algorithm: GEOPHYSICS, **63**, 1150–1165.

McCorduck, P., 2004, Machines who think: A personal inquiry into the history and prospects of artificial intelligence: AK Peters Ltd.

McHargue, T., M. Pyrcz, M. Sullivan, J. Clark, A. Fildani, B. Romans, J. Covault, M. Levy, H. Posamentier, and N. Drinkwater, 2010, Architecture of turbidite channel systems on the continental slope: Patterns and predictions: Mar Petroleum Geol, **28**.

Pham, N., S. Fomel, and D. Dunlap, 2018, *in* Automatic channel detection using deep learning: 2026–2030.

———, 2019, Automatic channel detection using deep learning: Interpretation, **7**, SE43–SE50.

Phillips, M., and S. Fomel, 2017, Plane-wave sobel attribute for discontinuity enhancement in seismic images: GEOPHYSICS, **82**, WB63–WB69.

Phillips, M., S. Fomel, and R. Swindeman, 2016, Structure-oriented plane-wave Sobel filter for edge detection in seismic images: 86th Annual International Meeting, SEG Expanded Abstracts, 1954–1959.

Prony, R., 1795, Essai expérimental et analytique: Annuaire de l'École Polytechnique, **1**, 24.

Randen, T., E. Monsen, C. Signer, A. Abrahamsen, J. O. Hansen, T. Sæter, J. Schlaf, and L. Sønneland, 2000, Three-dimensional texture attributes for seismic data analysis: 70th Annual International Meeting, SEG Expanded Abstracts, 668–671.

Redmon, J., S. K. Divvala, R. B. Girshick, and A. Farhadi, 2016, You only look once: Unified, real-time object detection: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788.

Ronneberger, O., P. Fischer, and T. Brox, 2015, U-net: Convolutional networks for biomedical image segmentation: CoRR, **abs/1505.04597**.

Rosleff-Soerensen, B., L. Reuning, S. Back, and P. Kukla, 2012, Seismic geomorphology and growth architecture of a miocene barrier reef, browse basin, nw-australia: Marine and Petroleum Geology, **29**, 233 – 254.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, 2015, ImageNet Large Scale Visual Recognition Challenge: International Journal of Computer Vision (IJCV), **115**, 211–252.

Salazar, M., L. Moscardelli, and L. Wood, 2016, Utilising clinoform architecture to understand the drivers of basin margin evolution: a case study in the taranaki basin, new zealand: Basin Research, **28**, 840–865.

Samuel, A. L., 1959, Some studies in machine learning using the game of checkers: IBM J. Res. Dev., **3**, 210–229.

Saxe, A. M., J. L. McClelland, and S. Ganguli, 2013, Exact solutions to the nonlinear dynamics of learning in deep linear neural networks: CoRR, **abs/1312.6120**.

Shelhamer, E., J. Long, and T. Darrell, 2017, Fully convolutional networks for semantic segmentation: IEEE Trans. Pattern Anal. Mach. Intell., **39**, 640–651.

Shi, Y., X. Wu, and S. Fomel, 2018, *in* Automatic salt-body classification using deep-convolutional neural network: 1971–1975.

Sobel, I., and G. Feldman, 1968, A 3x3 isotropic gradient operator for image processing: Stanford Artificial Project, 271–272.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014, Dropout: A simple way to prevent neural networks from overfitting: J. Mach. Learn. Res., **15**, 1929–1958.

Stockwell, R. G., L. Mansinha, and R. P. Lowe, 1996, Localization of the complex spectrum: the s transform: IEEE Transactions on Signal Processing, **44**, 998–1001.

Swindeman, R. L., 2015, Iterative seismic data interpolation using plane-wave shaping: M.S. thesis, The University of Texas at Austin.

Waldeland, A., and A. Solberg, 2017, Salt classification using deep learning: 79th conference and exhibition, eage: Extended Abstracts, https://doi. org/10.3997/2214-4609.201700918.

Wang, L., and J. M. Mendel, 1992, Adaptive minimum prediction-error deconvolution and source wavelet estimation using hopfield neural networks: GEOPHYSICS, **57**, 670–679.

Werbos, P. J., 1974, Beyond regression: New tools for prediction and analysis in the behavioral sciences: PhD thesis, Harvard University.

———, 1994, The roots of backpropagation: From ordered derivatives to neural networks and political forecasting: Wiley-Interscience.

Wu, X., 2017, Directional structure-tensor-based coherence to detect seismic faults and channels: GEOPHYSICS, **82**, A13–A17.

Wu, X., L. Liang, Y. Shi, and S. Fomel, 2019, Faultseg3d: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation: GEOPHYSICS, **84**, IM35–IM45.

Zhang, N., J. Donahue, R. Girshick, and T. Darrell, 2014, Part-based r-cnns for fine-grained category detection: Computer Vision – ECCV 2014, Springer International Publishing, 834–849.

Vita

Nam Pham grew up in Vungtau city, Vietnam. Nam attended University of Tulsa, Oklahoma to pursue a bachelor's degree. After graduating summa cum laude in 2017 with double degrees in Geophysics and Mathematics, Nam began his graduate studies in geophysics at the University of Texas at Austin in 2017. He is a member of the Texas Consortium for Computational Seismology at the Bureau of Economic Geology.

Email address: nam-pham@utexas.edu

This thesis was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.